

# **Managing Services and Faults in Oracle® Solaris 11.1**

Copyright © 1998, 2014, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée d'The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

# Contents

---

<b>Preface</b> .....	7
<b>1 Managing Services (Overview)</b> .....	9
About SMF in this Release .....	9
Introduction to SMF .....	10
Advantages to Using SMF .....	10
SMF Concepts .....	11
SMF Service .....	11
SMF Dependencies .....	12
Service Identifiers .....	12
Service States .....	13
SMF Manifests .....	14
SMF Profiles .....	14
Service Configuration Repository .....	15
SMF Administrative Layers .....	15
SMF Repository Backups .....	17
SMF Snapshots .....	17
SMF Service Error Logging .....	17
SMF Administrative and Programming Interfaces .....	18
SMF Command-Line Administrative Utilities .....	18
Service Management Configuration Library Interfaces .....	19
SMF Components .....	19
SMF Master Restarter Daemon .....	19
SMF Delegated Restarters .....	19
SMF Properties and Property Groups .....	20
Managing Information in the Service Configuration Repository .....	20
Viewing SMF Information .....	20
Modifying SMF Information .....	21

Deleting SMF Information .....	21
SMF and Booting .....	22
SMF Compatibility .....	22
Run Levels .....	23
When to Use Run Levels or Milestones .....	24
Determining a System's Run Level .....	24
/etc/inittab File .....	25
What Happens When the System Is Brought to Run Level 3 .....	26
<b>2 Managing Services (Tasks) .....</b>	<b>27</b>
Monitoring SMF Services .....	27
Monitoring Services (Task Map) .....	27
▼ How to List the Status of a Service .....	28
▼ How to List Customizations of a Service .....	29
▼ How to Show Which Services Are Dependent on a Service Instance .....	29
▼ How to Show Which Services a Service Is Dependent On .....	30
▼ How to Set Up Notification of SMF Transition Events .....	31
Managing SMF Services .....	32
Managing SMF Services (Task Map) .....	32
Using RBAC Rights Profiles With SMF .....	33
▼ How to Create an SMF Service .....	33
▼ How to Disable a Service Instance .....	34
▼ How to Enable a Service Instance .....	35
▼ How to Restart a Service .....	35
▼ How to Restore a Service That Is in the Maintenance State .....	36
▼ How to Create an SMF Profile .....	36
▼ How to Manually Create an SMF Profile .....	37
▼ How to Apply an SMF Profile .....	38
Configuring SMF Services .....	39
Configuring SMF Services (Task Map) .....	39
▼ How to Modify an SMF Service Property .....	39
▼ How to Modify Multiple Properties for One Service .....	40
▼ How to Modify a Service That Is Configured by a File .....	41
▼ How to Change an Environment Variable for a Service .....	41
▼ How to Change a Property for an inetd Controlled Service .....	42

▼ How to Delete Customizations for a Service .....	43
▼ How to Modify a Command-Line Argument for an inetd Controlled Service .....	44
▼ How to Convert inetd.conf Entries .....	45
Using Run Control Scripts .....	45
Using Run Control Scripts (Task Map) .....	45
▼ How to Use a Run Control Script to Stop or Start a Legacy Service .....	46
▼ How to Add a Run Control Script .....	47
▼ How to Disable a Run Control Script .....	47
▼ How to Convert a Run Control Script to an SMF Service .....	48
Troubleshooting the Service Management Facility .....	49
Troubleshooting SMF (Task Map) .....	49
▼ How to Debug a Service That Is Not Starting .....	50
▼ How to Repair a Corrupt Repository .....	50
▼ How to Boot Without Starting Any Services .....	52
▼ How to Boot in Verbose Mode .....	53
▼ How to Force an su login Prompt If the system/filesystem/local:default Service Fails During Boot .....	54
<b>3 Using the Fault Manager .....</b>	<b>55</b>
Fault Management Overview .....	55
Notification of Faults and Defects .....	57
Displaying Information About Faults or Defects .....	57
▼ How to Display Information About Faulty Components .....	58
▼ How to Identify Which CPUs Are Offline .....	60
▼ How to Display Information About Defective Services .....	60
Repairing Faults or Defects .....	61
fmadm replaced Command .....	62
fmadm repaired Command .....	62
fmadm acquit Command .....	62
Fault Management Log Files .....	63
Fault Statistics .....	63
<b>Index .....</b>	<b>65</b>



# Preface

---

*Managing Services and Faults in Oracle Solaris 11.1* is part of a documentation set that provides a significant portion of the Oracle Solaris system administration information. This guide focuses on the Oracle Solaris Service Management Facility (SMF) feature, as well as the Fault Management Architecture (FMA).

---

**Note** – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the *Oracle Solaris OS: Hardware Compatibility Lists*. This document cites any implementation differences between the platform types.

---

## Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P-1 Typographic Conventions

Typeface	Description	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name%</code> you have mail.
<b>AaBbCc123</b>	What you type, contrasted with onscreen computer output	<code>machine_name%</code> <b>su</b> Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <code>rm filename</code> .

TABLE P-1 Typographic Conventions (Continued)

Typeface	Description	Example
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> .  <i>A cache</i> is a copy that is stored locally.  Do <i>not</i> save the file.  <b>Note:</b> Some emphasized items appear bold online.

## Shell Prompts in Command Examples

The following table shows UNIX system prompts and superuser prompts for shells that are included in the Oracle Solaris OS. In command examples, the shell prompt indicates whether the command should be executed by a regular user or a user with privileges.

TABLE P-2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	\$
Bash shell, Korn shell, and Bourne shell for superuser	#
C shell	machine_name%
C shell for superuser	machine_name#



# Managing Services (Overview)

---

This chapter provides an overview of the Oracle Solaris Service Management Facility (SMF) feature. In addition, information about run levels is provided.

The following is a list of the information that is in this chapter:

- “Introduction to SMF” on page 10
- “SMF Concepts” on page 11
- “SMF Administrative and Programming Interfaces” on page 18
- “SMF Components” on page 19
- “SMF Compatibility” on page 22
- “Run Levels” on page 23
- “/etc/inittab File” on page 25

For information about the procedures associated with SMF, see “[Monitoring Services \(Task Map\)](#)” on page 27. For information about the procedures associated with run levels, see “[Using Run Control Scripts \(Task Map\)](#)” on page 45.

## About SMF in this Release

The following new SMF features have been added to this release:

- SMF property editing using the `svccfg editprop` command has been enhanced with the following improvements.
  - Displays the composed view when an instance is selected. It also displays where the property is defined, for example either in the instance or in the service
  - By default, the `editprop` subcommand does not display SMF infrastructure properties such as methods, dependencies, or SMF template definitions.

See “[How to Modify Multiple Properties for One Service](#)” on page 40 and the `svccfg(1M)` man page on your Oracle Solaris 11 Update 1 system for more information.

- The `svccprop` command was changed so that SMF template definitions are no longer displayed in its output. There is an `-a` option to include the template definitions if desired. See the `svccprop(1)` man page on your Oracle Solaris 11 Update 1 system for more information.
- The `svcbundle` command has been added to the release. This command allows you to create SMF manifests and profiles. See [“How to Create an SMF Service” on page 33](#), [“How to Create an SMF Profile” on page 36](#), and the `svcbundle(1M)` man page on your Oracle Solaris 11 Update 1 system for more information.

## Introduction to SMF

SMF makes it easier to manage applications and system services. The framework infrastructure augments the traditional UNIX startup scripts, `init` run levels, and configuration files. SMF provides a mechanism to define the relationships between applications or services, so that dependent services can automatically be restarted when necessary. Information needed to manage each service is stored in the service configuration repository, which provides a simplified way to manage each service.

SMF defines a set of actions that can be invoked on a service by an administrator. These actions, which can be manually manipulated by the `svcadm` command, include `enable`, `disable`, `refresh`, `restart`, and `mark`. Each service is managed by a service restarter, which carries out the administrative actions. In general, the restarters carry out actions by executing methods for a service. Methods for each service are defined in the service configuration repository. These methods allow the restarter to move the service from one state to another state.

The service configuration repository provides a per-service snapshot at the time that each service is successfully started so that fallback is possible. In addition, the repository provides a consistent and persistent way to enable or disable a service, as well as a consistent view of service states. This capability helps you debug service configuration problems.

## Advantages to Using SMF

The framework that SMF provides makes it easier to manage all of the services running on a server. In specific, the framework provides:

- Automatically restarts failed services in dependency order, whether they failed as the result of administrator error, software bug, or were affected by an uncorrectable hardware error. The dependency order is defined by dependency statements.
- Makes services objects that can be viewed, with the `svcs` command, and managed, with `svcadm` and `svccfg` commands. You can also view the relationships between services and processes using `svcs -p`, for both SMF services and legacy `init.d` scripts.

The `svcadm` command also allows for services to be enabled and disabled. These changes can persist through upgrades and reboots. If the `-t` option is used, the changes are temporary.

- Makes it easy to debug and ask questions about services by providing an explanation of why a service isn't running by using `svcs -x`. Also, this process is eased by individual and persistent log files for each service.
- Enhances the ability of administrators to securely delegate tasks to non-root users, including the ability to modify properties and enable, disable, or restart services on the system.
- Boots faster on large systems by starting services in parallel according to the dependencies of the services. The opposite process occurs during shutdown.
- Preserves compatibility with existing administrative practices wherever possible. For example, most customer and ISV-supplied rc scripts still work as usual.

## SMF Concepts

This section presents terms and their definitions within the SMF framework. These terms are used throughout the documentation. To grasp SMF concepts, an understanding of these terms is essential.

### SMF Service

The fundamental unit of administration in the SMF framework is the *service instance*. Each SMF service could be running multiple times on a system accounting for slightly different configurations. These different configurations are called service instances. Each *instance* is a specific configuration of a service. For example, a web server is a service. A specific web server daemon that is configured to listen on port 80 is an instance. Each instance of the web server service could have different configuration requirements. The service has system-wide configuration requirements, but each instance can override specific requirements, as needed. Multiple instances of a single service are managed as child objects of the service object.

Services are not just the representation for standard long-running system services such as `in.dhcpd` or `nfsd`. Services also represent varied system entities that include ISV applications. In addition, a service can represent less traditional entities such as the following:

- A physical network device
- A configured IP address
- Kernel configuration information
- Milestones that correspond to a system init state, such as the multiuser run level

Generically, a service is an entity that provides a list of capabilities to applications and other services, local and remote. A service is dependent on an implicitly and explicitly declared list of local services.

A *milestone* is a special type of service. Milestone services represent a level of system readiness. For example, run levels are represented by milestones in SMF. Also, milestones can be used to

indicate the readiness of a group of services, such as `svc:/milestone/name-services:default` for the name services or `svc:/milestone/config:default` for the `sysconfig` service.

## SMF Dependencies

*Dependencies* define the relationships between services. These relationships provide precise fault containment by restarting only those services that are directly affected by a fault, rather than restarting all of the services. Dependencies also provide a scalable and reproducible initialization process. Finally, the definition of precise dependencies allows system startup to take advantage of modern, highly parallel machines because all independent services can be started in parallel.

The restart behavior of a service is defined by the `restart_on` attribute for each dependency. A service can be configured to stop, if the service it is dependent on stops due to an error or for another reason, or is refreshed. After a service is stopped by this process, it will automatically be restarted as soon as the service it is dependent on starts. For example, the `ssh` service has a dependency on the `network/ipfilter` service. The `restart_on` attribute is set to `error`, which means that the `ssh` service will be stopped and automatically restarted if the `network/ipfilter` service stops due to an error. The `ssh` service will not be stopped if the other event types are encountered.

## Service Identifiers

Each service instance is named with a Fault Management Resource Identifier or FMRI. The FMRI includes the service name and the instance name. For example, the FMRI for the `rlogin` service is `svc:/network/login:rlogin`, where `network/login` identifies the service and `rlogin` identifies the service instance.

Equivalent formats for an FMRI are as follows:

- `svc://localhost/system/system-log:default`
- `svc:/system/system-log:default`
- `system/system-log:default`

In addition, many SMF commands may use an abbreviated service or instance name, when there is no ambiguity. For example, `system-log` can be used directly rather than the longer formats. See the SMF command man pages, such as `svcadm(1M)` or `svcs(1)`, for instructions on which FMRI formats are appropriate.

Service names include prefixes to help identify the purpose of each service. These prefixes include names such as `application`, `device`, `milestone`, `network`, or `system`. The `site` prefix is reserved for site-specific customizations, which means that a service named `svc:/site/service-name` will never conflict with the services delivered in an Oracle Solaris release.

Legacy `init.d` scripts are also represented with FMRI that start with `lrc` instead of `svc`, for example, `lrc:/etc/rc2_d/S47pppd`. The legacy service's initial start times during system boot are displayed by using the `svcs` command. However, you cannot administer these services by using SMF.

During initial system deployment, services listed in `/etc/inetd.conf` are automatically converted into SMF services. The FMRI for these services are slightly different. The syntax for a converted `inetd` service is:

```
network/service-name/protocol
```

In addition, the syntax for a converted service that uses the RPC protocol is:

```
network/rpc-service-name/rpc_protocol
```

Where *service-name* is the name defined in `/etc/inetd.conf` and *protocol* is the protocol for the service. The `inetconv` command can be used to convert `inetd.conf` entries after initial system deployment.

## Service States

The `svcs` command displays the state, start time, and FMRI of service instances. The state of each service is one of the following:

- `degraded` – The service instance is enabled, but is running at a limited capacity.
- `disabled` – The service instance is not enabled and is not running.
- `legacy_run` – The legacy service is not managed by SMF, but the service can be observed. This state is only used by legacy services.
- `maintenance` – The service instance has encountered an error that must be resolved by the administrator.
- `offline` – The service instance is enabled, but the service is not yet running or available to run.
- `online` – The service instance is enabled and has successfully started.
- `uninitialized` – This state is the initial state for all services before their configuration has been read.

An asterisk “\*” is appended to the state for instances in transition. A question mark “?” is displayed if the state is absent or unrecognized.

## SMF Manifests

An SMF *manifest* is an XML file that describes a service and a set of instances. Manifests are imported to load the properties of that service and its instances into the service configuration repository. See the [service\\_bundle\(4\)](#) man page for a complete description of the contents of an SMF manifest. Also, see the [svcbundle\(1M\)](#) man page for a description of a tool which makes creating manifests easier.

The preferred location for manifests is `/lib/svc/manifest`. Manifests stored there will be imported and upgraded by the `svc:/system/early-manifest-import:default` service during the boot process before any services start. Running the import process early ensures that the repository will contain information from the latest manifests before the services are started. At other times you can import information from these manifests by running this command: `svcadm restart manifest-import`. `/var/svc/manifest` remains available for compatibility purposes, but manifests located there will not be imported or upgraded until the `svc:/filesystem/minimal:default` instance is online, which indicates that `/var` is mounted.

Do not make changes to manifests delivered by Oracle or third-party software vendors. Do not directly edit those manifests in `/lib/svc/manifest` and `/var/svc/manifest`, as any customizations will be lost upon upgrade. Instead, either create a site profile to customize the service, or use the `svccfg` or `inetadm` command to manipulate the properties directly. The `/lib/svc/manifest/site` and `/var/svc/manifest/site` directories are also reserved for site-specific use. The Oracle Solaris release will not deliver manifests into those directories.

In the Oracle Solaris 11 release, multiple manifests can be used to describe a single service. This can be useful, for example, to define a new instance of a service without modifying the service's existing manifest. If the same property in the same administrative layer for the same service or instance is defined by multiple manifests, SMF cannot determine which value to use. When this type of conflict is detected, the instance is placed in the maintenance state. See “[SMF Administrative Layers](#)” on page 15 for a more information about layers.

## SMF Profiles

An SMF profile is an XML file that allows customization of services and instances that are delivered by the system. Profiles are available for customization by using a file rather than a set of scripts, or to customize the configuration at deployment or installation time.

All configurations can be customized by using a profile.

Local customizations must be placed in files named with a `.xml` suffix in the `/etc/svc/profile/site` directory. All customizations in this directory are applied when the system is booted or when the `svcadm restart manifest-import` command is run.

As with manifests, any conflicting definitions between files in `/etc/svc/profile/site` are treated as conflicts, and the affected instances are placed in the maintenance state.

A system profile is also applied during installation. Changes to the system profile in `/etc/svc/profile/generic.xml` are rarely necessary. See the [`smf\_bootstrap\(5\)`](#) man page for more information.

For more information about using profiles, see [“How to Apply an SMF Profile”](#) on page 38.

## Service Configuration Repository

The *service configuration repository* stores persistent configuration information as well as SMF runtime data for services. The repository is distributed among local memory and local files. The service configuration repository can only be manipulated or queried by using SMF interfaces. For more information about manipulating and accessing the repository, see the [`svccfg\(1M\)`](#) and [`svccprop\(1\)`](#) man pages. The service configuration repository daemon is covered in the [`svc.configd\(1M\)`](#) man page. The service configuration library is documented in the [`libscf\(3LIB\)`](#) man page.

Properties in the repository can be defined on either the service or the instance. Properties that are set on the service are shared by all instances of that service. Properties that are set on the instance are used only by that instance and can override properties on the service.

The `svccfg` command offers a *raw* view of properties, and is precise about whether the properties are set on the service or the instance. If you view a service by using the `svccfg` command, you cannot see instance properties. If you view the instance instead, you cannot see service properties. The `svccprop` command offers a *composed* view of the instance, where both instance properties and service properties are combined into a single property namespace. When service instances are started, the composed view of their properties is used.

All SMF configuration changes can be logged by using the Oracle Solaris auditing framework. Refer to [“Configuring the Audit Service \(Task Map\)”](#) in *Oracle Solaris 11.1 Administration: Security Services* for more information.

## SMF Administrative Layers

In the Oracle Solaris 11 release, information that records the source of properties, property groups, instances, and services has been added to the service configuration repository. This information enables users to determine which data are administrative customizations and which data were delivered with the software.

To help identify the source of an entity, the following layers are defined:

- The `admin` layer includes any changes that are made by using the SMF commands or by calling the `libscf(3LIB)` API.
- The `site-profile` layer includes any values from the files in the `/etc/svc/profile/site` directory or the legacy `/etc/svc/profile/site.xml` and `/var/svc/profile/site.xml` profiles.
- The `system-profile` layer includes any values from the system profile locations: `/etc/svc/profile/generic.xml` and `/etc/svc/profile/platform.xml`.
- The `manifest` layer includes values from a system manifest directory: `/lib/svc/manifest` or `/var/svc/manifest`.

To maintain compatibility for existing clients who expect a single property per property name, as well as to create a policy for overrides, the layering has a simple override behavior. The `admin` layer takes precedence. If a property has a value in the `admin` layer, that is the value that is used by the service. If not, the `site-profile` layer is checked, followed by the `system-profile` layer, and finally the `manifest` layer. This behavior allows for local customizations to take precedence over the values that are provided when the system was installed.

These layers are managed automatically by the system. An administrator's direct changes to the repository appear only in the `admin` layer. Other layers are changed only by placing or removing files in standard locations. When a property is placed into the repository due to file contents, the information about that property includes the name of the file that the contents came from.

An administrator cannot modify the lower layers directly by using `svccfg` or `libscf` calls. When the `svccfg delete`, `svccfg delpg`, or `svccfg delprop` command is used, the entity will be masked instead of fully deleted. Normally, users cannot see the deleted entity, but masked entities can be explicitly explored by using the `svccfg listcust` command, and unmasked by using the `svccfg delcust` command, if desired. Exploring masked entities allows administrators to see what a configuration would look like after the mask is removed, and make changes if necessary without harming the running system.

The `svccfg listprop` command has options to enable the exploration of these layers. For example, `svccfg listprop -l all` prints all layers and the values in each layer. In addition, the `svccfg listcust` command can be used to list customizations only.



## SMF Repository Backups

SMF automatically takes the following backups of the repository:

- The boot backup is taken immediately before the first change to the repository is made during each system startup.
- The `manifest_import` backups occur after `svc:/system/early-manifest-import:default` or `svc:/system/manifest-import:default` completes, if the service imported any new manifests or ran any upgrade scripts.

Four backups of each type are maintained by the system. The system deletes the oldest backup, when necessary. The backups are stored as `/etc/svc/repository-type-YYYYMMDD_HHMMSS`, where `YYYYMMDD` (year, month, day) and `HHMMSS` (hour, minute, second), are the date and time when the backup was taken. Note that the hour format is based on a 24-hour clock.

You can restore the repository from these backups, if an error occurs. To do so, use the `/lib/svc/bin/restore_repository` command. For more information, see [“How to Repair a Corrupt Repository” on page 50](#).

## SMF Snapshots

The data in the service configuration repository includes *snapshots*, as well as a configuration that can be edited. Data about each service instance is stored in the snapshots. The standard snapshots are as follows:

- `initial` – Taken on the first import of the manifest
- `running` – Taken when `svcadm refresh` is run
- `start` – Taken at the last successful start

The SMF service always executes with the `running` snapshot. This snapshot is automatically created if it does not exist.

The `svccfg` command is used to change current property values. Those values become visible to the service when the `svcadm refresh` command is run to integrate those values into the `running` snapshot. The `svccfg` command can also be used to view or revert to instance configurations in another snapshot.

## SMF Service Error Logging

Service-specific information, including errors the service or its methods emits, as well as information about enable actions, start times, and so on, are logged in individual files for each service instance in `/var/svc/log`. To determine the name of a service's log file, run the `svcs -x service` command.

By default, SMF writes log messages to the `syslog` program and the console only if administrative intervention is required, for example, if a service enters the maintenance state. Other options are available but rarely used. See the `svc.startd(1M)` man page for other potential configurations.

In addition, to error logging, the SMF service can be configured to notify you when an FMA event occurs or when services transition in to or out of a service state. These notifications can use the Simple Network Management Protocol (SNMP) or the Simple Mail Transfer Protocol (SMTP). See “[How to Set Up Notification of SMF Transition Events](#)” on page 31 for information about setting up SMF notifications.

## SMF Administrative and Programming Interfaces

This section introduces the interfaces that are available when you use SMF.

### SMF Command-Line Administrative Utilities

SMF provides a set of command-line utilities that interact with SMF and accomplish standard administrative tasks. The following utilities can be used to administer SMF.

TABLE 1-1 Service Management Facility Utilities

Command Name	Function
<code>svcadm</code>	Provides the ability to perform common service management tasks, such as enabling, disabling, or restarting service instances.
<code>svcbundle</code>	Generates new SMF manifests and profiles automatically.
<code>svccfg</code>	Provides the ability to directly display and manipulate the contents of the service configuration repository. Note: The contents displayed by this command are for the exact selected entity, not for the combined service and instance properties.
<code>svccprop</code>	Retrieves property values from the service configuration repository with an output format appropriate for use in shell scripts. Note: The contents displayed by this command are composed, and contains properties set on both the service and instance.
<code>svcs</code>	Gives detailed views of the service state of all service instances in the service configuration repository.
<code>inetadm</code>	Provides the ability to observe or configure services controlled by <code>inetd</code> .

## Service Management Configuration Library Interfaces

SMF provides a set of programming interfaces, the `libscf(3LIB)` API, that is used to interact with the service configuration repository through the `svc.configd` daemon. This daemon is the arbiter of all requests to the local repository datastores. A set of fundamental interfaces is defined as the lowest level of interaction possible with services in the service configuration repository. The interfaces provide access to all service configuration repository features such as transactions and snapshots.

Many developers only need a set of common tasks to interact with SMF. These tasks are implemented as convenience functions on top of the fundamental services to ease the implementation burden.

## SMF Components

SMF includes a master restarter daemon and delegated restarters. In addition, each service or service instance can store configuration data in properties. These properties are organized into property groups to make administration simpler.

### SMF Master Restarter Daemon

The `svc.startd` daemon is the master process starter and restarter. The daemon is responsible for managing service dependencies for the entire system. The daemon takes on the previous responsibility that `init` held of starting the appropriate `/etc/rc*.d` scripts at the appropriate run levels. First, `svc.startd` retrieves the information in the service configuration repository. Next, the daemon starts services when their dependencies are met. The daemon is also responsible for restarting services that have failed and for shutting down services whose dependencies are no longer satisfied. The daemon uses operating system events, such as process death, to keep track of service states.

### SMF Delegated Restarters

Some services have a set of common behaviors on startup. To provide commonality among these services, a delegated restarter might take responsibility for these services. In addition, a delegated restarter can be used to provide more complex or application-specific restarting behavior. The delegated restarter can support a different set of methods, but exports the same service states as the master restarter. The restarter's name is stored with the service. A current example of a delegated restarter is `inetd`, which can start Internet services on demand, rather than having the services always running. The defined restarter for each service can be displayed using the `svcs -l` command.

## SMF Properties and Property Groups

All information in the service configuration repository is stored as a set of properties, which are grouped by property groups. *Property groups* are used to group different types of service information. Property groups can be applied to the system as well as to individual services. Some of the common system-defined property groups include:

- `general` – Contains information about the service instance, including the `general/enabled` property, which defines whether the instance is enabled
- `restarter` – Contains runtime information that is stored by the service's restarter, including the `restarter/state` property, which shows the current state of the service
- `start` – Contains the start method definition, including the `start/exec` property, which defines what program to execute to start the service

Another common property group is `config`. The `config` property group is not system defined, so it can be used by services developers to hold service data. See the [smf\(5\)](#) man page for more information about properties and property groups.

## Managing Information in the Service Configuration Repository

There are many ways to view, modify, or delete information by using SMF. This section discusses which methods are best for which uses.

### Viewing SMF Information

You can use the `svccfg` and `svccprop` commands to view information in the service configuration repository. For a complete description of these commands, see the [svccfg\(1M\)](#) and [svccprop\(1\)](#) man pages.

- `svccprop` – Lists the values assigned to property groups or properties in running snapshot. Because this command combines data for the service, and service instance, it provides a comprehensive or “composed” view of the data.
- `svccfg listpg` – Lists information about property groups in the selected service, service instance, or snapshot. All property group names, types, and flags are listed.
- `svccfg listprop` – Lists information about properties and property group in the selected service, service instance, or snapshot. For property groups, the names, types and flags are listed. For properties, the names, types and values are listed.
  - `-l layer_name` – Lists the properties and property groups within the named layer in a service or service instance. Using `all` as a layer name lists all of the layers and properties for that service. Note that each service and service instance is displayed separately.

- -f – Lists the file name that a property came from.
- -o – Selects the fields to display.
- `svccfg listcust` – Lists any site customizations in the `site-profile` or `admin` layer. Also listed are any masked entries for the selected service or service instance.
  - -M – Lists only masked entities.
  - -L – Shows all local customizations, which includes both administrative customizations and site profile customizations.

## Modifying SMF Information

You can modify information about a service or service instance by using a manifest, or a profile, or by using the `svccfg` command. Any changes you make with the `svccfg` command are recorded in the `admin` layer. The following list includes some of the options that you can use to modify information by using the `svccfg` command:

- `addpg` – Adds a property group to the selected service or service instance
- `addpropvalue` – Assigns an additional value to a existing property
- `setenv` – Sets an environment variable for a service or service instance
- `setprop` – Sets the value of a named property in the selected service
- `setnotify` – Sets notification parameters for software events and FMA event classes

## Deleting SMF Information

You can delete information about a service or service instance by using the `svccfg` command. The following list includes some of the options that you can use to remove information by using the `svccfg` command:

- `delcust` – Deletes any administrative customizations for the selected service
- `delpropvalue` – Deletes all property values that match the given string
- `unsetenv` – Removes an environment variable for a service or service instance

When you delete information from the repository one of two things happen, either the entity will be masked or the entity will be removed. Any information that has been defined only in the `admin` layer will be removed. Any information from a manifest or profile will be masked, so that the standard commands will not display the information. The information is masked so that if you need to undo the deletion, you readily will have the information available.

## SMF and Booting

When a system is being booted, you can select the milestone to boot to or the level of error messages to be recorded as follows:

- You can choose a specific milestone to boot to by using this command:

```
ok boot -m milestone=milestone
```

The default is `all`, which starts all enabled services. Also useful is `none`, which starts only `init`, `svc.startd`, and `svc.configd`, and provides a debugging environment where services can be started manually. See [“How to Boot Without Starting Any Services” on page 52](#) for instructions on how to use the `none` milestone.

The run level equivalents `single-user`, `multi-user`, and `multi-user-server` are also available, but are not commonly used. In particular, `multi-user-server` does not start any services that are not dependent on that milestone, so might not include important services.

- You can choose the level of logging for `svc.startd` by using this command:

```
ok boot -m logging=level
```

The logging levels that you can select include `quiet` and `verbose`. See [“SMF Service Error Logging” on page 17](#) for specific information about the logging levels.

## SMF Compatibility

Although many standard services are now managed by SMF, the scripts placed in `/etc/rc*.d` continue to be executed on run level transitions. Most of the `/etc/rc*.d` scripts that were included in previous releases have been removed as part of SMF. The ability to continue to run the remaining scripts allows for third-party applications to be added without having to convert the services to use SMF.

In addition, `/etc/inittab` entries also continue to be processed by the `init` command. Also, `/etc/inetd.conf` is available for packages to amend. During initial system deployment, services that are listed in `/etc/inetd.conf` are automatically converted into SMF services. Any later additions can be converted by using the `inetconv` command. The status of these services can be viewed, but no other changes are supported through SMF. Applications that use this conversion feature will not benefit from the precise fault containment provided by SMF. The latest version of `inetd` does not look for entries in `/etc/inetd.conf` to convert after the initial boot.

Applications that are converted to utilize SMF no longer need to make use of the mechanisms listed in this section.

## Run Levels

A system's *run level* (also known as an *init state*) defines what services and resources are available to users. A system can be in only one run level at a time.

The release has eight run levels, which are described in the following table. The default run level is specified in the `/etc/inittab` file as run level 3.

TABLE 1–2 Oracle Solaris Run Levels

Run Level	Init State	Type	Purpose
0	Power-down state	Power-down	To shut down the operating system so that it is safe to turn off power to the system.
s or S	Single-user state	Single-user	To run as a single user with some file systems mounted and accessible.
1	Administrative state	Single-user	To access all available file systems. User logins are disabled.
2	Multiuser state	Multiuser	For normal operations. Multiple users can access the system and all file system. All daemons are running except for the NFS and SMB server daemons.
3	Multiuser level with NFS resources shared	Multiuser	For normal operations with NFS and SMB resources shared. This is the default run level.
4	Alternative multiuser state	Multiuser	Not configured by default, but available for customer use.
5	Power-down state	Power-down	To shut down the operating system so that it is safe to turn off power to the system. If possible, automatically turns off power on systems that support this feature.
6	Reboot state	Reboot	To shut down the system to run level 0, and then reboot to multiuser level with NFS and SMB resources shared (or whatever level is the default in the <code>inittab</code> file).

In addition, the `svcadm` command can be used to change the run level of a system, by selecting a milestone at which to run. The following table shows which run level corresponds to each milestone.

TABLE 1-3 Run Levels and SMF Milestones

Run Level	SMF Milestone FMRI
S	milestone/single-user:default
2	milestone/multi-user:default
3	milestone/multi-user-server:default

## When to Use Run Levels or Milestones

In general, changing milestones or run levels is an uncommon procedure. If it is necessary, using the `init` command to change to a run level will change the milestone as well and is the appropriate command to use. The `init` command is also useful for shutting down a system.

However, booting a system by using the `none` milestone can be very useful when you are debugging startup problems. There is no equivalent run level to the `none` milestone. See [“How to Boot Without Starting Any Services” on page 52](#) for specific instructions.

## Determining a System's Run Level

Display run level information by using the `who -r` command.

```
$ who -r
```

Use the `who -r` command to determine a system's current run level for any level.

### EXAMPLE 1-1 Determining a System's Run Level

This example displays information about a system's current run level and previous run levels.

```
$ who -r
.   run-level 3  Dec 13 10:10  3  0  S
$
```

Output of <code>who -r</code> command	Description
run-level 3	Identifies the current run level
Dec 13 10:10	Identifies the date of last run level change
3	Also identifies the current run level
0	Identifies the number of times the system has been at this run level since the last reboot
S	Identifies the previous run level



## /etc/inittab File

When you boot the system or change run levels with the `init` or `shutdown` command, the `init` daemon starts processes by reading information from the `/etc/inittab` file. This file defines these important items for the `init` process:

- That the `init` process will restart
- What processes to start, monitor, and restart if they terminate
- What actions to take when the system enters a new run level

Each entry in the `/etc/inittab` file has the following fields:

*id*:*rstate*:*action*:*process*

The following table describes the fields in an `inittab` entry.

TABLE 1-4 Fields Descriptions for the `inittab` File

Field	Description
<i>id</i>	Is a unique identifier for the entry.
<i>rstate</i>	Lists the run levels to which this entry applies.
<i>action</i>	Identifies how the process that is specified in the process field is to be run. Possible values include: <code>sysinit</code> , <code>boot</code> , <code>bootwait</code> , <code>wait</code> , and <code>respawn</code> . For a description of the other action keywords, see <a href="#">inittab(4)</a> .
<i>process</i>	Defines the command or script to execute.

EXAMPLE 1-2 Default `inittab` File

The following example shows a default `inittab` file that is installed with the release. A description for each line of output in this example follows.

```
ap::sysinit:/usr/sbin/autopush -f /etc/iu.ap          (1)
smf::sysinit:/lib/svc/bin/svc.startd    >/dev/msglog 2<>/dev/msglog </dev/console    (2)
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/...    (3)
```

1. Initializes STREAMS modules
2. Initializes the master restarter for SMF
3. Describes a power fail shutdown

## What Happens When the System Is Brought to Run Level 3

1. The `init` process is started and reads the properties that are defined in the `svc:/system/environment:init` SMF service to set any environment variables.
2. The `init` process reads the `inittab` file and does the following:
  - a. Executes any process entries that have `sysinit` in the action field so that any special initializations can take place before users login
  - b. Passes the startup activities to `svc.startd`

For a detailed description of how the `init` process uses the `inittab` file, see the [init\(1M\)](#) man page.

# Managing Services (Tasks)

---

This chapter covers the tasks required to manage and monitor the Service Management Facility (SMF). In addition, information about managing run level scripts is provided. The following topics are covered:

- “Monitoring SMF Services” on page 27
- “Managing SMF Services” on page 32
- “Configuring SMF Services” on page 39
- “Using Run Control Scripts” on page 45
- “Troubleshooting the Service Management Facility” on page 49

## Monitoring SMF Services

The following tasks show how to monitor SMF services.

### Monitoring Services (Task Map)

The following task map describes the procedures that are needed to monitor SMF services.

Task	Description	For Instructions
Display the status of a service instance.	Displays the status of all running service instances.	“How to List the Status of a Service” on page 28
Display the customizations to a service.	Displays the customizations in the admin layer for the service.	“How to List Customizations of a Service” on page 29
Display the service dependents.	Display the services that are dependent on the specified service.	“How to Show Which Services Are Dependent on a Service Instance” on page 29

Task	Description	For Instructions
Display the dependencies of a service.	Display the services that a specified service is dependent on. This information can be used to help identify what is preventing a service from starting.	<a href="#">“How to Show Which Services a Service Is Dependent On” on page 30</a>
Configure notification for significant events.	Uses SNMP or SMTP to send a notification of a change in service state or of an FMA event.	<a href="#">“How to Set Up Notification of SMF Transition Events” on page 31</a>

## ▼ How to List the Status of a Service

This procedure can be used to show what services are running.

- **Run the `svcs` command.**

Running this command without any options displays a status report of the service specified by the FMRI.

```
$ svcs -l FMRI
```

### Example 2-1 Showing the Status of the `sendmail` Service

This example shows the status of a service that includes dependencies. Also, the `-p` option is used so that information about the process ID, start time and command run is displayed

```
$ svcs -lp network/smtp:sendmail
fmri      svc:/network/smtp:sendmail
name     sendmail SMTP mail transfer agent
enabled  true
state    online
next_state none
state_time Tue Aug 09 19:25:54 2011
logfile  /var/svc/log/network-smtp:sendmail.log
restarter svc:/system/svc/restarter:default
contract_id 114
manifest  /etc/svc/profile/generic_limited_net.xml
manifest  /lib/svc/manifest/network/smtp-sendmail.xml
dependency require_all/refresh file:///localhost/etc/mail/sendmail.cf (online)
dependency require_all/refresh file:///localhost/etc/nsswitch.conf (online)
dependency optional_all/none svc:/system/filesystem/autofs (online)
dependency require_all/none svc:/system/filesystem/local (online)
dependency require_all/none svc:/network/service (online)
dependency require_all/refresh svc:/milestone/name-services (online)
dependency optional_all/refresh svc:/system/identity:domain (online)
dependency optional_all/none svc:/system/system-log (online)
process  101077 /usr/lib/sendmail -bd -q15m
```

**Example 2-2** Showing the Status of all Services

The following command lists all services that are installed on the system as well as the status of each service. The command displays those services that are disabled as well as those that are enabled.

```
$ svcs -a
```

**Example 2-3** Showing all Notification Parameters

The following command lists all of the FMA events notification parameters and the SMF state transition parameters.

```
$ svcs -n
```

**Example 2-4** Showing the Status of Services Controlled by `inetd`

The following command lists services that are controlled by `inetd`. Each service's FMRI is listed, along with the run state and whether the service is enabled or disabled.

```
$ inetadm
```

## ▼ How to List Customizations of a Service

- List local customizations.

This command displays all of the changes at the `admin` layer for the selected service.

```
% /usr/sbin/svccfg -s FMRI listcust
```

## ▼ How to Show Which Services Are Dependent on a Service Instance

This procedure shows how to determine which service instances depend on the specified service.

- Display the service dependents.

```
$ svcs -D FMRI
```

**Example 2-5** Displaying the Service Instances That Are Dependent on the `multiuser` Milestone

The following example shows how to determine which service instances are dependent on the `multiuser` milestone.

```
$ svcs -D milestone/multi-user
STATE      STIME    FMRI
disabled   Aug_09   svc:/application/time-slider:default
disabled   Aug_09   svc:/application/management/net-snmp:default
online     Aug_09   svc:/system/intrd:default
online     Aug_09   svc:/system/boot-config:default
online     Aug_09   svc:/milestone/multi-user-server:default
```

## ▼ How to Show Which Services a Service Is Dependent On

This procedure shows how to determine which services a specified service instance is dependent on.

- Display the service dependencies.

```
$ svcs -d FMRI
```

### Example 2-6 Displaying the Service Instances That the multiuser Milestone Is Dependent On

The following example shows the services instances that the multiuser milestone is dependent on:

```
$ svcs -d milestone/multi-user:default
STATE      STIME    FMRI
disabled   Aug_09   svc:/network/nfs/client:default
disabled   Aug_09   svc:/network/smb/client:default
disabled   Aug_09   svc:/system/mdmonitor:default
disabled   Aug_09   svc:/application/print/server:default
online     Aug_09   svc:/system/resource-controls:default
online     Aug_09   svc:/system/metasync:default
online     Aug_09   svc:/system/rmtmpfiles:default
online     Aug_09   svc:/system/utmp:default
online     Aug_09   svc:/system/name-service/cache:default
online     Aug_09   svc:/system/device/audio:default
online     Aug_09   svc:/network/rpc/bind:default
online     Aug_09   svc:/milestone/name-services:default
online     Aug_09   svc:/network/iscsi/initiator:default
online     Aug_09   svc:/milestone/single-user:default
online     Aug_09   svc:/system/filesystem/local:default
online     Aug_09   svc:/network/inetd:default
online     Aug_09   svc:/system/cron:default
online     Aug_09   svc:/system/filesystem/autofs:default
online     Aug_09   svc:/system/filesystem/ufs/quota:default
online     Aug_09   svc:/system/power:default
online     Aug_09   svc:/system/system-log:default
online     Aug_09   svc:/system/system-log:default
online     Aug_09   svc:/system/auditd:default
online     Aug_09   svc:/network/smtp/sendmail
```

## ▼ How to Set Up Notification of SMF Transition Events

This procedure causes the system to generate an email notification each time one of the services or a selected service has a change in state. You can choose to use either SMTP or SNMP. Normally, you would only select SNMP if you already have SNMP configured for some other reason.

By default, SNMP traps are sent on maintenance transitions. If you use SNMP for monitoring, you can configure additional traps for other state transitions.

### 1 Become an administrator or assume a role that includes the Service Management rights profile.

For more information, see [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: Security Services](#).

### 2 Set notification parameters.

The following examples show how to set notification parameters for SMF and FMA events, as well as how to list and delete notification parameters.

#### Example 2-7 Configuring Notifications for All SMF Service State Events

The following command creates a notification that sends email when services go into the maintenance state.

```
# /usr/sbin/svccfg setnotify -g maintenance mailto:sysadmins@example.com
```

You can also choose to select services that start in the state listed with the `from` option and end in the state listed with the `to` option. The valid SMF states for this option are: `degraded`, `maintenance`, `offline`, and `online`. You can use the `-g all` option to generate email for all state transition events. See the Notification Parameters section in the [smf\(5\)](#) man page for more information.

#### Example 2-8 Configuring Notifications for an Individual Service

The following command creates a notification that sends email when the `switch` service goes into the `online` state.

```
# /usr/sbin/svccfg -s svc:/system/name-service/switch:default setnotify to-online \
    mailto: sysadmins@example.com
```

#### Example 2-9 Configuring Notifications for FMA Events

The following command creates a notification that sends an SNMP message when a FMA problem is repaired.

```
# /usr/sbin/svccfg setnotify problem-repaired snmp:
```

The FMA event classes include `problem-diagnosed`, `problem-updated`, `problem-repaired` and `problem-resolved`. See the Notification Parameters section in the `smf(5)` man page for more information.

**Example 2-10 Listing Notification Settings**

The following command shows the notification settings for a new problem diagnosed by the FMA service. Notification settings for SMF service state transition events can be displayed by including the service state instead of the event class or by not including any arguments with `listnotify`.

```
# /usr/sbin/svccfg listnotify problem-diagnosed
```

**Example 2-11 Deleting Notification Settings**

The following command deletes the notification settings associated with the `switch` service transitioning to the `online` service state. You can use an FMA event class in place of the service state.

```
# /usr/sbin/svccfg -s svc:/system/name-service/switch:default delnotify to-online
```

## Managing SMF Services

This section includes information on managing SMF services.

### Managing SMF Services (Task Map)

The following task map describes the procedures that are needed to manage SMF services.

Task	Description	For Instructions
Create a service.	Allows for a script to be managed through the SMF service.	<a href="#">“How to Create an SMF Service” on page 33</a>
Disable a service instance.	Stops a running service and prevents the service from restarting.	<a href="#">“How to Disable a Service Instance” on page 34</a>
Enable a service instance.	Starts a service. In addition, the service will be restarted during subsequent reboots.	<a href="#">“How to Enable a Service Instance” on page 35</a>
Restart a service.	Restarts a service without having to use separate commands to disable and then enable the service.	<a href="#">“How to Restart a Service” on page 35</a>
Restore a service in the maintenance state.	Shows how to clean up and restart a service that is in the maintenance state.	<a href="#">“How to Restore a Service That Is in the Maintenance State” on page 36</a>



Task	Description	For Instructions
Create a profile.	Create a profile to easily deploy customizations without running the <code>svccfg</code> command to make all of the changes.	<a href="#">“How to Create an SMF Profile” on page 36</a>
	Manually create a profile to easily deploy customizations.	<a href="#">“How to Manually Create an SMF Profile” on page 37</a>
Apply a profile.	Uses the information in a profile to disable, enable, or customize services as needed.	<a href="#">“How to Apply an SMF Profile” on page 38</a>

## Using RBAC Rights Profiles With SMF

You can use RBAC rights profiles to allow users to manage some of the SMF services, without having to give the user root access. The rights profiles define what commands the user can run. For SMF, the following profiles have been created:

- **Service Management:** User can add, delete or modify services.
- **Service Operator:** User can request state changes of any service instance, such as restart and refresh.

For specific information about the authorizations, see the `smf_security(5)` man page. For instructions to assign a rights profile, see [“How to Change the Security Attributes of a User” in \*Oracle Solaris 11.1 Administration: Security Services\*](#).

### ▼ How to Create an SMF Service

Administrators often write scripts to do work when the system is booting. This procedure shows how to use the advantages of SMF to support a script.

**Before You Begin** Determine which service model the script needs. If the script does some work and then exits without starting any long running processes, it is most likely a transient service. If the script starts up a long running daemon and then exits, then it is a contract service. See the Duration section in the `svc.startd(1M)` man page for more information about service models.

- 1 Become an administrator or assume a role that includes the Service Management rights profile.**  
For more information, see [“How to Use Your Assigned Administrative Rights” in \*Oracle Solaris 11.1 Administration: Security Services\*](#).
- 2 Copy the start method script to `/Lib/svc/method/ex_svc`.**

**3 Create a temporary manifest.**

In this example, the service name is `site/ex_svc` and it is a transient service so it doesn't need a stop method.

```
# svcbundle -o /tmp/ex_svc.xml -s service-name=site/ex_svc \  
-s start-method=/lib/svc/method/ex_svc -s model=transient
```

**4 Make any additional changes to the manifest.**

Verify that the contents of manifest in `/tmp/ex_svc.xml` is what you need.

**5 Copy the manifest to the correct directory.**

```
# cp /tmp/ex_svc.xml /lib/svc/manifest/site/ex_svc.xml
```

**6 Import the manifest and start the service.**

```
# svcadm restart svc:/system/manifest-import
```

**Example 2–12 Automatically Installing a Generated Manifest**

To automatically install the manifest in `/lib/svc/manifest/site`, use the `-i` option with the `svcbundle` command as show below. This command can be run in place of the command in step 3, and you will not need to do any other steps.

```
# svcbundle -i -s service-name=site/ex_svc \  
-s start-method=/lib/svc/method/ex_svc -s model=transient
```

## ▼ How to Disable a Service Instance

Use the following procedure to disable a service. The service status change is recorded in the service configuration repository. Once the service is disabled, the disabled state will persist across reboots. The only way to get the service running again is to enable it.

**1 Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: Security Services](#).

**2 Check the dependents of the service you want to disable.**

If this service has dependents that you need, then you cannot disable this service.

```
# svcs -D FMRI
```

**3 Disable the service.**

```
# svcadm disable FMRI
```

## ▼ How to Enable a Service Instance

Use the following procedure to enable a service. The service status change is recorded in the service configuration repository. After the service is enabled, the enabled state will persist across system reboots, however the service will start only if all dependencies are met.

- 1 **Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see “[How to Use Your Assigned Administrative Rights](#)” in *Oracle Solaris 11.1 Administration: Security Services*.

- 2 **Enable a service.**

```
# svcadm enable FMRI
```

- 3 **Verify that the service has been enabled.**

```
# svcs -x FMRI
```

- 4 **Enable service dependencies.**

If required dependencies are disabled, enable them with the following command:

```
# svcadm enable -r FMRI
```

### Example 2–13 Enabling a Service in Single-user Mode

The following command enables `rpcbind`. The `-t` option starts the service in temporary mode which does not change the service repository, so this change will not persist across a reboot. The `-r` option recursively starts all the dependencies of the named service.

```
# svcadm enable -rt rpc/bind
```

## ▼ How to Restart a Service

If a service is currently running but needs to be restarted due to a configuration change or some other reason, the service can be restarted without you having to type separate commands to stop and start the service. The only reason to specifically disable and then enable a service is if changes need to be made before the service is enabled, and after the service is disabled.

- 1 **Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see “[How to Use Your Assigned Administrative Rights](#)” in *Oracle Solaris 11.1 Administration: Security Services*.

- 2 **Restart a service.**

```
# svcadm restart FMRI
```

## ▼ How to Restore a Service That Is in the Maintenance State

- 1 **Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

- 2 **Determine why the service is in maintenance.**

```
# svcs -x FMRI
```

Consult the log file or man page mentioned to determine what the error is.

- 3 **Determine if any process that are dependent to the service have not stopped.**

Normally, when a service instance is in a maintenance state, all processes associated with that instance have stopped. However, you should make sure before you proceed. The following command lists the contract ID for the selected FMRI.

```
# svcs -o CTID FMRI
```

- 4 **(Optional) Kill any remaining processes.**

Repeat this step for all processes that are displayed by the svcs command.

```
# pkill -9 -c CTID
```

- 5 **Restore the service.**

```
# svcadm clear FMRI
```

## ▼ How to Create an SMF Profile

A profile is an XML file which lists SMF services and whether each should be enabled or disabled. Profiles are used to enable or disable many services at once. Profiles are also used to set property values, add property values, and even create a service and instances of a service. Not all services need to be listed in a profile. Each profile only needs to include those services that need to be enabled or disabled to make the profile useful.

The `svc:/system/rmtmpfiles` service is responsible for cleaning up the `/tmp` directory on boot. By default the `rmtmpfiles` service does not cleanup `/var/tmp`. If you wish to clean up `/var/tmp` during the boot process, you can change the behavior by setting the `options/clean_vartmp` property to `true`. If you want to do this on multiple systems, the easiest thing to do is to create a profile and place it in `/etc/svc/profile/site` on those systems.

- 1 **Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

**2 Create the profile.**

This command creates a new profile in `/tmp/rmtmpfiles.xml`, which can then be copied to `/etc/svc/profile/site` as needed.

```
# svcbundle -o /tmp/rmtmpfiles.xml -s service-name=system/rmtmpfiles \
-s bundle-type=profile -s service-property=options:clean_vartmp:boolean:true
```

**3 Copy the profile to the correct directory.**

```
# cp /tmp/rmtmpfiles.xml /etc/svc/profile/site/rmtmpfiles.xml
```

**4 Restart the manifest-import service to apply the profile to the system.**

```
# svcadm restart svc:/system/manifest-import
```

**Example 2–14 Automatically Installing an SMF Profile While Creating the Profile**

The `-i` option is used to install the generated profile. The command will write the profile to `/etc/svc/profile/site/rmtmpfiles.xml`. Also, the command will restart the `manifest-import` service to apply the profile.

```
# svcbundle -i -s service-name=system/rmtmpfiles \
-s bundle-type=profile -s service-property=options:clean_vartmp:boolean:true
```

**▼ How to Manually Create an SMF Profile**

A profile is an XML file which lists SMF services and whether each should be enabled or disabled. Profiles are used to enable or disable many services at once. Profiles are also used to set property values, add property values, and even create a service and instances of a service. Not all services need to be listed in a profile. Each profile only needs to include those services that need to be enabled or disabled to make the profile useful.

**1 Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see [“How to Use Your Assigned Administrative Rights” in \*Oracle Solaris 11.1 Administration: Security Services\*](#).

**2 Create a profile.**

In this example, the `svccfg` command is used to create a profile that represents the local customizations on the current system. Alternatively, you could make a copy of an existing profile to edit.

```
# svccfg extract> profile.xml
```

If you have large numbers of identical systems, or if you want to archive the system configuration for later restoration, you might want to use this procedure to create a unique version of a SMF profile.

### 3 Edit the `profile.xml` file to make any required changes.

#### a. Change the name of the profile in the `service_bundle` declaration.

In this example the name is changed to `profile`.

```
# cat profile.xml
...
<service_bundle type='profile' name='profile'
  xmlns:xi='http://www.w3.org/2003/XInclude'
  ...
```

#### b. Remove any services that should not be managed by this profile.

For each service, remove the three lines that describe the service. Each service description starts with `<service` and ends with `</service>`. This example shows the lines for the LDAP client service.

```
# cat profile.xml
...
<service name='network/ldap/client' version='1' type='service'>
  <instance name='default' enabled='true' />
</service>
```

#### c. Add any services that should be managed by this profile.

Each service needs to be defined using the three line syntax shown above.

#### d. If necessary, change the enabled flag for selected services.

In this example, the `sendmail` service is disabled.

```
# cat profile.xml
...
<service name='network/smtp' version='1' type='service'>
  <instance name='sendmail' enabled='false' />
</service>
...
```

### 4 When necessary, apply the new profile.

See [“How to Apply an SMF Profile” on page 38](#) for instructions.

## ▼ How to Apply an SMF Profile

### 1 Become an administrator or assume a role that includes the Service Management rights profile.

For more information, see [“How to Use Your Assigned Administrative Rights” in \*Oracle Solaris 11.1 Administration: Security Services\*](#).

### 2 Apply a profile.

In this example, the `profile.xml` profile is applied.

```
# svccfg apply profile.xml
```

# Configuring SMF Services

The following tasks show how to configure SMF services. In particular, how to modify service properties and other configuration information for a service or a service instance.

## Configuring SMF Services (Task Map)

The following task map describes the procedures that are needed to configure SMF services.

Task	Description	For Instructions
Modify a service.	<p>Modifies a service property of a specified service instance.</p> <p>Modifies multiple properties of a specified service instance.</p> <p>Modifies the configuration file of a service instance.</p> <p>Modifies an environment variable of a service instance.</p> <p>Deletes the customizations in the <code>admin</code> layer for a service.</p>	<p><a href="#">“How to Modify an SMF Service Property” on page 39</a></p> <p><a href="#">“How to Modify Multiple Properties for One Service” on page 40</a></p> <p><a href="#">“How to Modify a Service That Is Configured by a File” on page 41</a></p> <p><a href="#">“How to Change an Environment Variable for a Service” on page 41</a></p> <p><a href="#">“How to Delete Customizations for a Service” on page 43</a></p>
Modify an <code>inetd</code> service.	<p>Changes a configuration property of a service controlled by <code>inetd</code>.</p> <p>Changes the startup options of a service controlled by <code>inetd</code>.</p>	<p><a href="#">“How to Change a Property for an <code>inetd</code> Controlled Service” on page 42</a></p> <p><a href="#">“How to Modify a Command-Line Argument for an <code>inetd</code> Controlled Service” on page 44</a></p>
Convert <code>inetd.conf</code> entries.	Converts <code>inetd</code> services into legacy-run services that can be monitored using SMF.	<a href="#">“How to Convert <code>inetd.conf</code> Entries” on page 45</a>

### ▼ How to Modify an SMF Service Property

This procedure shows how to modify the property that identifies the user that can start a service.

#### 1 Become an administrator or assume a role that includes the Service Management rights profile.

For more information, see [“How to Use Your Assigned Administrative Rights” in \*Oracle Solaris 11.1 Administration: Security Services\*](#).

**2 Change the value that is assigned to the start/user property.**

First, give the FMRI to identify the appropriate service. Next, assign the UID that will start the service.

```
# svccfg -s FMRI
svc:/service: setprop start/user = astring: newlogin
```

**3 Refresh the service.**

```
# svcadm refresh FMRI
```

## ▼ How to Modify Multiple Properties for One Service

This procedure shows how to modify several service properties without having to type in as much information as the `setprop` subcommand requires. In this example, several DNS service properties are changed.

**1 Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

**2 Change the property values.**

The `editprop` subcommand will gather the property groups and properties of specified service into a temporary file and the program named by the `VISUAL` or the `EDITOR` environment variable is invoked to edit it. See the `svccfg(1M)` man page for more information.

```
# svccfg -s network/dns/client editprop

##
## The value used to construct the "nameserver" directive in resolv.conf(4).
##
setprop config/nameserver = net_address: (192.168.1.10 192.168.1.11)

##
## The value used to construct the "search" directive in resolv.conf(4).
##
# setprop config/search = astring: "example.com sales.example.com"
```

Do not save the file at this point.

**3 Apply the changes to all instances.**

Uncomment the refresh line at the bottom of the file to apply these changes when you save the file and exit the editor.

```
## Uncomment to apply these changes to all instances of this service.
refresh
```



## ▼ How to Modify a Service That Is Configured by a File

The following procedure shows how to change the configuration of a service that is not managed by the `inetd` service.

**1 Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: Security Services](#).

**2 Make changes to the configuration files, as needed.**

Many of the services have one or more configuration files that are used to define the startup or other configuration information. These files can be changed while the service is running. The contents of the files is only checked when the service is started.

**3 Restart the service.**

```
# svcadm restart FMRI
```

### Example 2–15 Adding a New NTP Server

To add a new NTP server to support your NTP clients, add a new entry for the server to the `/etc/inet/ntp.conf` file. Next, restart the NTP service. This example shows you what the `ntp.conf` file could look like, as well as how to restart the service.

```
# cat /etc/inet/ntp.conf
.
.
server ntpserver1.example.com
server ntpserver2.example.com
# svcadm restart svc:/network/ntp:default
```

## ▼ How to Change an Environment Variable for a Service

This procedure shows how to modify `cron` environment variables to help with debugging.

**1 Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: Security Services](#).

**2 Verify that the service is running.**

```
# svcs system/cron
STATE          STIME      FMRI
online         Dec_04     svc:/system/cron:default
```

**3 Set environment variables.**

In this example the UMEM\_DEBUG and LD\_PRELOAD environment variables are set. For information about the setenv subcommand refer to the [svccfg\(1M\)](#) man page.

```
# svccfg -s system/cron:default setenv UMEM_DEBUG default
# svccfg -s system/cron:default setenv LD_PRELOAD libumem.so
```

**4 Refresh and restart the service.**

```
# svcadm refresh system/cron
# svcadm restart system/cron
```

**5 Verify that the change has been made.**

```
# pargs -e 'pgrep -f /usr/sbin/cron'
100657: /usr/sbin/cron
envp[0]: LOGNAME=root
envp[1]: LD_PRELOAD=libumem.so
envp[2]: PATH=/usr/sbin:/usr/bin
envp[3]: SMF_FMRI=svc:/system/cron:default
envp[4]: SMF_METHOD=/lib/svc/method/svc-cron
envp[5]: SMF_RESTARTER=svc:/system/svc/restarter:default
envp[6]: TZ=GB
envp[7]: UMEM_DEBUG=default
```

## ▼ How to Change a Property for an inetd Controlled Service

**1 Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see “[How to Use Your Assigned Administrative Rights](#)” in *Oracle Solaris 11.1 Administration: Security Services*.

**2 List the properties for the specific service.**

This command displays all of the properties for the service identified by the FMRI.

```
# inetadm -l FMRI
```

**3 Change the property for the service.**

Each property for an inetd controlled service is defined by a property name and an assigned value. Supplying the property name without a specified value resets the property to the default value. Specific information about the properties for a service should be covered in the man page associated with the service.

```
# inetadm -m FMRI property-name=value
```

**4 Verify that the property has changed.**

List the properties again to make sure that the appropriate change has occurred.

```
# inetadm -l FMRI
```

## 5 Confirm that the change has taken effect.

Confirm the property change that the change has the desired effect.

### Example 2-16 Changing the `tcp_trace` Property for `telnet`

The following example shows how to set the `tcp_trace` property for `telnet` to `true`. Checking the `syslog` output after running a `telnet` command shows that the change has taken effect.

```
# inetadm -l svc:/network/telnet:default
SCOPE      NAME=VALUE
           name="telnet"
.
.
default   inherit_env=TRUE
default   tcp_trace=FALSE
default   tcp_wrappers=FALSE
# inetadm -m svc:/network/telnet:default tcp_trace=TRUE
# inetadm -l svc:/network/telnet:default
SCOPE      NAME=VALUE
           name="telnet"
.
.
default   inherit_env=TRUE
           tcp_trace=TRUE
default   tcp_wrappers=FALSE
# telnet localhost
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.
login: root
Password:
.
.
.
Last login: Mon Jun 21 05:55:45 on console
Sun Microsystems Inc. SunOS 5.10 s10_57 May 2004
# ^D
Connection to localhost closed by foreign host.
# tail -1 /var/adm/messages
Jun 21 06:04:57 yellow-19 inetd[100308]: [ID 317013 daemon.notice] telnet[100625]
from 127.0.0.1 32802
```

## ▼ How to Delete Customizations for a Service

Customizations can be removed from properties, property groups, and service instances.

### ● Delete local customizations.

This command deletes all of the changes at the admin layer for the selected service.

```
% /usr/sbin/svccfg -s FMRI delcust
```

**Example 2–17** Removing Customizations from a Property

This example shows the effect of deleting customizations from a service. First we apply a customization to set the `clean_vartmp` property in the `rmtmpfiles` service so that `/var/tmp` gets cleaned during the boot process. As a privileged user type, the following commands:

```
# svccfg -s svc:/system/rmtmpfiles setprop options/clean_vartmp = true
# svcadm refresh svc:/system/rmtmpfiles
```

Next verify that the property has been changed:

```
$ svcprop -p options/clean_vartmp svc:/system/rmtmpfiles
true
```

To observe the customization, type the following commands:

```
$ svccfg -s svc:/system/rmtmpfiles listcust
options/clean_vartmp    boolean    admin    true
$ svcprop -p options/clean_vartmp -l all svc:/system/rmtmpfiles
options/clean_vartmp boolean manifest false
options/clean_vartmp boolean admin true
```

To remove the customization, type the following command as a privileged user:

```
# svccfg -s svc:/system/rmtmpfiles delcust options/clean_vartmp
Deleting customizations for property: options/clean_vartmp
```

Now when you run the commands to observe the customization, you see:

```
$ svccfg -s svc:/system/rmtmpfiles listcust
$ svcprop -p options/clean_vartmp -l all svc:/system/rmtmpfiles
options/clean_vartmp boolean manifest false
```

## ▼ How to Modify a Command-Line Argument for an inetd Controlled Service

- 1 **Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: Security Services](#).

- 2 **List the `exec` property for the specific service.**

This command displays all the properties for the service identified by the FMRI. Adding the `grep` command restricts the output to the `exec` property for the service.

```
# inetadm -l FMRI|grep exec
```

**3 Change the `exec` property for the service.**

The `command-syntax` set with the `exec` property defines the command string that is run when the service is started.

```
# inetadm -m FMRI exec="command-syntax"
```

**4 Verify that the property has changed.**

List the properties again to make sure that the appropriate change has occurred.

```
# inetadm -l FMRI
```

## ▼ How to Convert `inetd.conf` Entries

The following procedure converts `inetd.conf` entries into SMF service manifests. This procedure needs to be run any time a third-party application that depends on `inetd` is added to a system. Also run this procedure, if you need to make configuration changes to the entry in `/etc/inetd.conf`.

**1 Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: Security Services](#).

**2 Convert the `inetd.conf` entries.**

The `inetconv` command converts each entry in the selected file into service manifests.

```
# inetconv -i filename
```

### Example 2-18 Converting `/etc/inet/inetd.conf` Entries into SMF Service Manifests

```
# inetconv -i /etc/inet/inetd.conf
```

## Using Run Control Scripts

The following procedures show how to use run control scripts to stop or start a legacy service. Also included are instructions for adding or removing a run control script.

### Using Run Control Scripts (Task Map)

The following task map includes several procedures that are associated with using run control scripts. Each row includes a task, a description of when you would want to perform that task, followed by a link to the task.

Task	Description	For Instructions
Stop or start a legacy service.	Use a run control script to stop or start a service.	“How to Use a Run Control Script to Stop or Start a Legacy Service” on page 46
Add a run control script.	Create a run control script and add it to the <code>/etc/init.d</code> directory.	“How to Add a Run Control Script” on page 47
Disable a run control script.	Disable a run control script by renaming the file.	“How to Disable a Run Control Script” on page 47
Convert a run control script to use SMF.	Replaces the run control script with an SMF manifest so that the service can be managed by the SMF service.	“How to Convert a Run Control Script to an SMF Service” on page 48

## ▼ How to Use a Run Control Script to Stop or Start a Legacy Service

- 1 **Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

- 2 **Stop the system service.**

```
# /etc/init.d/filename stop
```

- 3 **Restart the system service.**

```
# /etc/init.d/filename start
```

- 4 **Verify that the service has been stopped or started.**

```
# pgrep -f service
```

### Example 2–19 Using a Run Control Script to Stop or Start a Service

For example, you can stop the PPPoE server daemon by typing the following:

```
# /etc/init.d/pppd stop
# pgrep -f nfs
```

Then, you can restart the PPPoE server daemon by typing the following:

```
# /etc/init.d/pppd start
# pgrep -f ppp
101748
# pgrep -f ppp -d, | xargs ps -fp
  UID    PID  PPID  C  STIME TTY          TIME CMD
 daemon 101748    1   0   Sep 01 ?        0:06 /usr/lib/inet/pppoed
```

## ▼ How to Add a Run Control Script

If you want to add a run control script to start and stop a service, copy the script into the `/etc/init.d` directory. Then, create links in the `rcn.d` directory where you want the service to start and stop.

See the README file in each `/etc/rcn.d` directory for more information on naming run control scripts. The following procedure describes how to add a run control script.

**1 Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

**2 Add the script to the `/etc/init.d` directory.**

```
# cp filename/etc/init.d
# chmod 0744 /etc/init.d/filename
# chown root:sys /etc/init.d/filename
```

**3 Create links to the appropriate `rcn.d` directory.**

```
# cd /etc/init.d
# ln filename /etc/rc2.d/Snnfilename
# ln filename /etc/rcn.d/Knnfilename
```

**4 Verify that the script has links in the specified directories.**

```
# ls /etc/init.d/*filename /etc/rc2.d/*filename /etc/rcn.d/*filename
```

### Example 2–20 Adding a Run Control Script

The following example shows how to add a run control script for the xyz service.

```
# cp xyz /etc/init.d
# chmod 0744 /etc/init.d/xyz
# chown root:sys /etc/init.d/xyz
# cd /etc/init.d
# ln xyz /etc/rc2.d/S99xyz
# ln xyz /etc/rc0.d/K99xyz
# ls /etc/init.d/*xyz /etc/rc2.d/*xyz /etc/rc0.d/*xyz
```

## ▼ How to Disable a Run Control Script

You can disable a run control script by renaming it with an underscore (`_`) at the beginning of the file name. Files that begin with an underscore or dot are not executed. If you copy a file by adding a suffix to it, both files will be run.

- 1 **Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: Security Services.](#)

- 2 **Rename the script by adding an underscore ( \_ ) to the beginning of the new file.**

```
# cd /etc/rcn.d
# mv filename_filename
```

- 3 **Verify that the script has been renamed.**

```
# ls _*
_filename
```

### Example 2–21 Disabling a Run Control Script

The following example shows how to rename the S99datainit script.

```
# cd /etc/rc2.d
# mv S99datainit _S99datainit
# ls _*
_S99datainit
```

## ▼ How to Convert a Run Control Script to an SMF Service

This procedure shows how to use the advantages of SMF to support an existing run control script. See the `svcbundle(1M)` man page for more information.

- Before You Begin** First, determine which service model the script needs. If the script does some work and then exits without starting any long running processes, it is most likely a transient service. If the script starts up a long running daemon and then exits, then it is a contract service. See the Duration section in the `svc.startd(1M)` man page for more information about service models.

Next, determine which run level is appropriate for the service. This can be determined by looking at which directory the script is linked into, for example `/etc/rc3.d`.

- 1 **Become an administrator or assume a role that includes the Service Management rights profile.**

For more information, see [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: Security Services.](#)

- 2 **Create a temporary manifest.**

In this example, the service name is `ex_svc`. This is a contract service and runs at level 3.

```
# svcbundle -o /tmp/ex_svc.xml -s service-name=ex_svc
-s start-method=/lib/svc/method/ex_svc -s model=transient
```

- 3 **Make any additional changes to the manifest.**

Verify that the contents of manifest in `/tmp/ex_svc.xml` is what you need.



- 4 **Copy the manifest to the correct directory.**  
`# cp /tmp/ex_svc.xml /lib/svc/manifest/site/ex_svc.xml`
- 5 **Stop the existing service.**  
`# /etc/init.d/ex_svc stop`
- 6 **Disable the run control script.**  
 Remove any links to the run control script from the appropriate `rcn.d` directories.
- 7 **Import the manifest and start the service.**  
`# svcadm restart svc:/system/manifest-import`

## Troubleshooting the Service Management Facility

The following procedures show how to troubleshoot or fix SMF services. Some of these procedures also show how to change boot parameters to alter the way a system boots.

### Troubleshooting SMF (Task Map)

The following task map includes several procedures that can be used to troubleshoot problems on your system. Each row includes a task, a description of when you would want to perform that task, followed by a link to the task.

Task	Description	For Instructions
Debug a service that is not running	Gather information from the SMF service about why a service isn't running.	<a href="#">“How to Debug a Service That Is Not Starting” on page 50</a>
Repair a corrupt repository	Steps to follow to replace a corrupt repository so that the system will boot completely.	<a href="#">“How to Repair a Corrupt Repository” on page 50</a>
Boot without starting any services	Boot a system without any running services for debugging a system that hangs during the boot process.	<a href="#">“How to Boot Without Starting Any Services” on page 52</a>
Generate more boot messages	Steps to create more messages during the boot process to aid with debugging.	<a href="#">“How to Boot in Verbose Mode” on page 53</a>
Stop the boot process if local file systems are not available	Steps to get the login service started so that you can investigate the problem, if local file systems are not mounted.	<a href="#">“How to Force an suLogin Prompt If the system/filesystem/local:default Service Fails During Boot” on page 54</a>

## ▼ How to Debug a Service That Is Not Starting

In this procedure, the print service is disabled.

### 1 Become an administrator or assume a role that includes the Service Management rights profile.

For more information, see “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

### 2 Request information about the service that is not running due to an error.

```
# svcs -xv
svc:/application/print/server:default (LP Print Service)
  State: disabled since Wed 13 Oct 2004 02:20:37 PM PDT
Reason: Disabled by an administrator.
  See: http://support.oracle.com/msg/SMF-8000-05
  See: man -M /usr/share/man -s 1M lpsched
Impact: 2 services are not running:
  svc:/application/print/rfc1179:default
  svc:/application/print/ipp-listener:default
```

The -x option provides additional information about the service instances that are impacted.

### 3 Enable the service.

```
# svcadm enable application/print/server
```

## ▼ How to Repair a Corrupt Repository

This procedure shows how to replace a corrupt repository with a default copy of the repository. When the repository daemon, `svc.configd`, is started, it does an integrity check of the configuration repository. This repository is stored in `/etc/svc/repository.db`. The repository can become corrupted due to one of the following reasons:

- Disk failure
- Hardware bug
- Software bug
- Accidental overwrite of the file

If the integrity check fails, the `svc.configd` daemon writes a message to the console similar to the following:

```
svc.configd: smf(5) database integrity check of:
```

```
  /etc/svc/repository.db
```

```
failed. The database might be damaged or a media error might have
prevented it from being verified. Additional information useful to
your service provider is in:
```

```
  /system/volatile/db_errors
```

The system will not be able to boot until you have restored a working database. `svc.startd(1M)` will provide a `sulogin(1M)` prompt for recovery purposes. The command:

```
/lib/svc/bin/restore_repository
```

can be run to restore a backup version of your repository. See <http://support.oracle.com/msg/SMF-8000-MY> for more information.

The `svc.startd` daemon then exits and starts `sulogin` to enable you to perform maintenance.

### 1 Enter the root password at the suLogin prompt.

The `sulogin` command enables the root user to enter system maintenance mode to repair the system.

### 2 Run the following command:

```
# /lib/svc/bin/restore_repository
```

Running this command takes you through the necessary steps to restore a non-corrupt backup. SMF automatically takes backups of the repository at key system moments. For more information see “[SMF Repository Backups](#)” on page 17.

When started, the `/lib/svc/bin/restore_repository` command displays a message similar to the following:

```
See http://support.oracle.com/msg/SMF-8000-MY for more information on the use of
this script to restore backup copies of the smf(5) repository.
```

```
If there are any problems which need human intervention, this script will
give instructions and then exit back to your shell.
```

After the root ( / ) file system is mounted with write permissions, or if the system is a local zone, you are prompted to select the repository backup to restore:

```
The following backups of /etc/svc/repository.db exists, from
oldest to newest:
```

```
... list of backups ...
```

Backups are given names, based on type and the time the backup was taken. Backups beginning with `boot` are completed before the first change is made to the repository after system boot.

Backups beginning with `manifest_import` are completed after `svc:/system/manifest-import:default` finishes its process. The time of the backup is given in `YYYYMMDD_HHMMSS` format.

### 3 Enter the appropriate response.

Typically, the most recent backup option is selected.

Please enter either a specific backup repository from the above list to restore it, or one of the following choices:

CHOICE	ACTION
boot	restore the most recent post-boot backup
manifest_import	restore the most recent manifest_import backup
-seed-	restore the initial starting repository (All customizations will be lost, including those made by the install/upgrade process.)
-quit-	cancel script and quit

Enter response [boot]:

If you press Enter without specifying a backup to restore, the default response, enclosed in [] is selected. Selecting -quit- exits the restore\_repository script, returning you to your shell prompt.

---

**Note** – Selecting -seed- restores the seed repository. This repository is designed for use during initial installation and upgrades. Using the seed repository for recovery purposes should be a last resort.

---

After the backup to restore has been selected, it is validated and its integrity is checked. If there are any problems, the restore\_repository command prints error messages and prompts you for another selection. Once a valid backup is selected, the following information is printed, and you are prompted for final confirmation.

After confirmation, the following steps will be taken:

```

svc.startd(1M) and svc.configd(1M) will be quiesced, if running.
/etc/svc/repository.db
  -- renamed --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS
/system/volatile/db_errors
  -- copied --> /etc/svc/repository.db_old_YYYYMMDD_HHMMSS_errors
repository_to_restore
  -- copied --> /etc/svc/repository.db
and the system will be rebooted with reboot(1M).

```

Proceed [yes/no]?

#### 4 Type yes to remedy the fault.

The system reboots after the restore\_repository command executes all of the listed actions.

## ▼ How to Boot Without Starting Any Services

If problems with starting services occur, sometimes a system will hang during the boot. This procedure shows how to troubleshoot this problem.

**1 Boot without starting any services.**

This command instructs the `svc.startd` daemon to temporarily disable all services and start `su` login on the console.

```
ok boot -m milestone=none
```

**2 Log in to the system as root.****3 Enable all services.**

```
# svcadm milestone all
```

**4 Determine where the boot process is hanging.**

When the boot process hangs, determine which services are not running by running `svcs -a`. Look for error messages in the log files in `/var/svc/log`.

**5 After fixing the problems, verify that all services have started.****a. Verify that all needed services are online.**

```
# svcs -x
```

**b. Verify that the `console-login` service dependencies are satisfied.**

This command verifies that the `login` process on the console will run.

```
# svcs -l system/console-login:default
```

**6 Continue the normal booting process.****▼ How to Boot in Verbose Mode**

By default, the level of messages that are displayed during a boot is set to the quiet mode, which generates messages when there is an error when a service starts. When troubleshooting a problem that occurs when a system boots, you may want to see more of the messages. This procedure shows you how to boot the system so that all of the error messages are displayed.

**● Boot in verbose mode.**

```
# boot -m verbose
```

**Example 2–22 Storing Boot Messages in a Log**

Giving the `boot` command the `-m` debug option causes all per-service and service startup messages to be logged in the log files.

## ▼ How to Force an suLogin Prompt If the system/filesystem/local:default Service Fails During Boot

Local file systems that are not required to boot the system are mounted by the `svc:/system/filesystem/local:default` service. When any of those file systems are unable to be mounted, the service enters a maintenance state. System startup continues, and any services which do not depend on `filesystem/local` are started. Services which require `filesystem/local` to be online before starting through dependencies are not started.

To change the configuration of the system so that a `suLogin` prompt appears immediately after the service fails instead of allowing system startup to continue, follow the procedure below.

### 1 Modify the system/console-login service.

```
# svccfg -s svc:/system/console-login
svc:/system/console-login> addpg site,filesystem-local dependency
svc:/system/console-login> setprop site,filesystem-local/entities = fmri: svc:/system/filesystem/local

svc:/system/console-login> setprop site,filesystem-local/grouping = astring: require_all

svc:/system/console-login> setprop site,filesystem-local/restart_on = astring: none

svc:/system/console-login> setprop site,filesystem-local/type = astring: service

svc:/system/console-login> end
```

### 2 Refresh the service.

```
# svcadm refresh console-login
```

**Troubleshooting** When a failure occurs with the `system/filesystem/local:default` service, the `svcs -vx` command should be used to identify the failure. After the failure has been fixed, the following command clears the error state and allows the system boot to continue: `svcadm clear filesystem/local`.

## Using the Fault Manager

---

The Oracle Solaris OS includes an architecture for building and deploying systems and services that are capable of predictive self healing. The service that is the core of the Fault Management Architecture (FMA) receives data related to hardware and software errors, automatically diagnoses the underlying problem, and responds by trying to take faulty components offline.

The following is a list of the information that is in this chapter:

- “Fault Management Overview” on page 55
- “Notification of Faults and Defects” on page 57
- “Displaying Information About Faults or Defects” on page 57
- “Repairing Faults or Defects” on page 61
- “Fault Management Log Files” on page 63
- “Fault Statistics” on page 63

### Fault Management Overview

The Oracle Solaris Fault Management feature provides an architecture for building resilient error handlers, structured error telemetry, automated diagnostic software, response agents, and structured messaging. Many parts of the software stack participate in Fault Management, including the CPU, memory and I/O subsystems, Oracle Solaris ZFS, an increasing set of device drivers, and other management stacks.

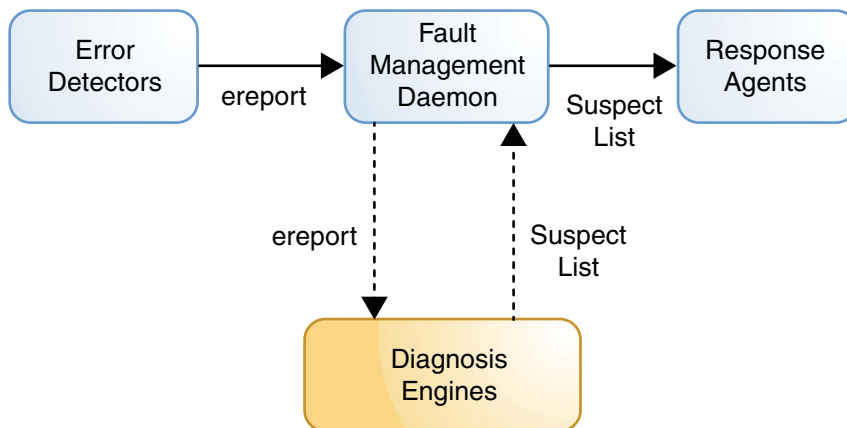
FMA is intended to help with problems that can occur on an Oracle Solaris system. The problem could be a fault, meaning that something that used to work but no longer does. The problem could be a defect, meaning that it never worked correctly. In general, hardware can experience both faults and defects. However, most software problems are defects or are caused by configuration issues.

At a high level, the Fault Management stack contains error detectors, diagnosis engines, and response agents. *Error detectors*, as the name suggests, detect errors in the system and perform any immediate, required handling. Error detectors issue well-defined error reports, or *ereports*,

to a diagnosis engine. A *diagnosis engine* interprets ereports and determines whether a fault or defect is present in the system. When such a determination is made, the diagnosis engine issues a *suspect list* that describes the resource or set of resources that might be the cause of the problem. The resource might or might not have an associated field-replaceable unit (FRU), a label, or an Automatic System Reconfiguration Unit (ASRU). An ASRU may be immediately removed from service to mitigate the problem until the FRU is replaced.

When the suspect list includes multiple suspects, for example, if the diagnosis engine cannot isolate a single suspect, the suspects are assigned a probability as to each suspect being the key suspect. The probabilities in this list add up to 100 percent. Suspect lists are interpreted by response agents. A *response agent* attempts to take some action based on the suspect list. Responses include logging messages, taking CPU strands offline, retiring memory pages, and retiring I/O devices.

Error detectors, diagnosis engines, and response agents are connected by the Fault Manager daemon, `fmd`, which acts as a multiplexor between the various components, as shown in the following figure.



The Fault Manager daemon is itself a service under SMF control. The service is enabled by default and controlled just like any other SMF service. See the `smf(5)` man page for more information.

The FMA and SMF services interact with each other when appropriate. Certain hardware problems can cause services to be stopped or restarted by SMF. Also, certain SMF errors cause FMA to report a defect.



## Notification of Faults and Defects

Often, the first interaction with the Fault Manager daemon is a system message indicating that a fault or defect has been diagnosed. Messages are sent to both the console and the `/var/adm/messages` file. All messages from the Fault Manager daemon use the following format:

```

1   SUNW-MSG-ID: SUN4V-8001-8H, TYPE: Fault, VER: 1, SEVERITY: Minor
2   EVENT-TIME: Wed Aug 24 21:56:03 UTC 2011
3   PLATFORM: SUNW,T5440, CSN: -, HOSTNAME: bur419-61
4   SOURCE: cpumem-diagnosis, REV: 1.7
5   EVENT-ID: 7b83c87c-78f6-6a8e-fa2b-d0cf16834049
6   DESC: The number of integer register errors associated with this thread has
7   exceeded acceptable levels.
8   AUTO-RESPONSE: The fault manager will attempt to remove the affected thread
9   from service.
10  IMPACT: System performance may be affected.
11  REC-ACTION: Use 'fmadm faulty' to provide a more detailed view of this
12  event. Please refer to the associated reference document at
13  http://support.oracle.com/msg/SUN4V-8001-8H for the latest service procedures and
14  policies regarding this diagnosis.
```

When notified of a diagnosed problem, always consult the recommended knowledge article for additional details. See line 13 above for an example. The knowledge article might contain additional actions that you or a service provider should take beyond those listed on line 11.

Notification of Fault Manager error events can be configured by using the Simple Network Management Protocol (SNMP) or the Simple Mail Transfer Protocol (SMTP). See [“How to Set Up Notification of SMF Transition Events” on page 31](#) for instructions.

In addition, Oracle Auto Service Request can be configured to automatically request Oracle service when specific hardware problems occur. See the [Oracle Auto Service Request product page](#) for information about this feature. The documentation link on this page provides links to *Oracle ASR Quick Installation Guide* and *Oracle ASR Installation and Operations Guide*.

## Displaying Information About Faults or Defects

The preferred method to display fault or defect information and determine the FRUs involved is the `fmadm faulty` command. However, the `fmdump` command is also supported. `fmdump` is often used to display a historical log of problems on the system, and `fmadm faulty` is used to display the active problems.




---

**Caution** – Do not base administrative action on the output of the `fmdump` command, but rather on the `fmadm faulty` output. The log files can contain error statements, which should not be considered faults or defects.

---

## ▼ How to Display Information About Faulty Components

### 1 Become an administrator.

For more information, see “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

### 2 Display information about the components.

```
# fmadm faulty
```

See the following examples for a description of the text generated.

#### Example 3-1 fmadm Output With One Faulty CPU

```

1  # fmadm faulty
2  -----
3  TIME                EVENT-ID                MSG-ID                SEVERITY
4  -----
5  Aug 24 17:56:03 7b83c87c-78f6-6a8e-fa2b-d0cf16834049  SUN4V-8001-8H  Minor
6
7  Host      : bur419-61
8  Platform  : SUNW,T5440          Chassis_id  : BEL07524BN
9  Product_sn : BEL07524BN
10
11 Fault class : fault.cpu.ultraSPARC-T2plus.ireg
12 Affects    : cpu:///cpuid=0/serial=1F95806CD1421929
13           : faulted and taken out of service
14 FRU       : "MB/CPU0" (hc:///product-id=SUNW,T5440:server-id=bur419-61:\
15           : serial=3529:part=541255304/motherboard=0/cpuboard=0)
16           : faulty
17 Serial ID. : 3529
18           : 1F95806CD1421929
19
20 Description : The number of integer register errors associated with this thread
21           : has exceeded acceptable levels.
22
23 Response   : The fault manager will attempt to remove the affected thread from
24           : service.
25
26 Impact    : System performance may be affected.
27
28 Action     : Use 'fmadm faulty' to provide a more detailed view of this event.
29           : Please refer to the associated reference document at
30           : http://support.oracle.com/msg/SUN4V-8001-8H for the latest service
31           : procedures and policies regarding this diagnosis.
```

Of primary interest is line 14, which shows the data for the impacted FRUs. The more human-readable location string is presented in quotation marks, "MB/CPU0". The quoted value is intended to match the label on the physical hardware. The FRU is also represented in a Fault Management Resource Identifier (FMRI) format, which includes descriptive properties about the system containing the fault, such as its host name and chassis serial number. On platforms that support it, the part number and serial number of the FRU are also included in the FRU's FMRI.

The Affects lines (lines 12 and 13) indicate the components that are affected by the fault and their relative state. In this example, a single CPU strand is affected. It is faulted and taken out of service.

Following the FRU description in the `fmadm faulty` command output, line 16 shows the state as `faulty`. The Action section might also include other specific actions instead of, or in addition to, the usual reference to the `fmadm` command.

### Example 3-2 `fmadm` Output With Multiple Faults

```

1  # fmadm faulty
2  -----
3  TIME                EVENT-ID                MSG-ID                SEVERITY
4  -----
5  Sep 21 10:01:36    d482f935-5c8f-e9ab-9f25-d0aaafec1e6c  PCIEX-8000-5Y  Major
6
7  Fault class   : fault.io.pci.device-invreq
8  Affects      : dev:///pci@0,0/pci1022,7458@11/pci1000,3060@0
9                dev:///pci@0,0/pci1022,7458@11/pci1000,3060@1
10               ok and in service
11               dev:///pci@0,0/pci1022,7458@11/pci1000,3060@2
12               dev:///pci@0,0/pci1022,7458@11/pci1000,3060@3
13               faulty and taken out of service
14  FRU          : "SLOT 2" (hc:///.../pciexrc=3/pciexbus=4/pciexdev=0)
15               repair attempted
16               "SLOT 3" (hc:///.../pciexrc=3/pciexbus=4/pciexdev=1)
17               acquitted
18               "SLOT 4" (hc:///.../pciexrc=3/pciexbus=4/pciexdev=2)
19               not present
20               "SLOT 5" (hc:///.../pciexrc=3/pciexbus=4/pciexdev=3)
21               faulty
22
23  Description   : The transmitting device sent an invalid request.
24
25  Response     : One or more device instances may be disabled
26
27  Impact      : Possible loss of services provided by the device instances
28               associated with this fault
29
30  Action       : Use 'fmadm faulty' to provide a more detailed view of this event.
31               Please refer to the associated reference document at
32               http://support.oracle.com/msg/PCIEX-8000-5Y for the latest service
33               procedures and policies regarding this diagnosis.

```

Following the FRU description in the `fmadm faulty` command output, line 21 shows the state as `faulty`. Other state values that you might see in other situations include `acquitted` and `repair attempted`, as shown for SLOT 2 and SLOT 3 in lines 15 and 17.

### Example 3-3 Showing Faults with the `fmddump` Command

Some console messages and knowledge articles might instruct you to use the older `fmddump -v -u UUID` command to display fault information. Although the `fmadm faulty` command is preferred, the `fmddump` command still operates, as shown in the following example:

```

1 % fmdump -v -u 7b83c87c-78f6-6a8e-fa2b-d0cf16834049
2 TIME          UUID          SUNW-MSG-ID EVENT
3 Aug 24 17:56:03.4596 7b83c87c-78f6-6a8e-fa2b-d0cf16834049 SUN4V-8001-8H Diagnosed
4 100% fault.cpu.ultraSPARC-T2plus.ireg
5
6 Problem in: -
7 Affects: cpu:///cpuid=0/serial=1F95806CD1421929
8 FRU: hc://:product-id=SUNW,T5440:server-id=bur419-61:\
9 serial=9999:part=541255304/motherboard=0/cpuid=0
10 Location: MB/CPU0

```

The information about the affected FRUs is still present, although separated across three lines (lines 8 through 10). The Location string presents the human-readable FRU string. The FRU lines presents the formal FMRI. Note that the severity, descriptive text, and action are not shown with the `fmdump` command, unless you use the `-m` option. See the [fmdump\(1M\)](#) man page for more information.

## ▼ How to Identify Which CPUs Are Offline

- Display information about the CPUs.

```

% /usr/sbin/psrinfo
0 faulted since 05/13/2011 12:55:26
1 on-line since 05/12/2011 11:47:26

```

The `faulted` state indicates that the CPU has been taken offline by a Fault Management response agent.

## ▼ How to Display Information About Defective Services

- 1 Become an administrator.

For more information, see [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: Security Services](#).

- 2 Display information about the defect.

```

# fmadm faulty
-----
TIME          EVENT-ID          MSG-ID          SEVERITY
-----
May 12 22:52:47 915cb64b-e16b-4f49-efe6-de81ff96fce7 SMF-8000-YX    major

Host          : parity
Platform     : Sun-Fire-V40z   Chassis_id   : XG051535088
Product_sn   : XG051535088

Fault class  : defect.sunos.smf.svc.maintenance
Affects     : svc:///system/intrd:default
              faulted and taken out of service
Problem in  : svc:///system/intrd:default

```

faulted and taken out of service

Description : A service failed - it is restarting too quickly.

Response : The service has been placed into the maintenance state.

Impact : svc:/system/intrd:default is unavailable.

Action : Run 'svcs -xv svc:/system/intrd:default' to determine the generic reason why the service failed, the location of any logfiles, and a list of other services impacted. Please refer to the associated reference document at <http://support.oracle.com/msg/SMF-8000-YX> for the latest service procedures and policies regarding this diagnosis.

### 3 Display information about the defective service.

Follow the instructions given in the Action section in the `fmadm` output.

```
# svcs -xv svc:/system/intrd:default
svc:/system/intrd:default (interrupt balancer)
  State: maintenance since Wed May 12 22:52:47 2010
  Reason: Restarting too quickly.
    See: http://support.oracle.com/msg/SMF-8000-YX
    See: man -M /usr/share/man -s 1M intrd
    See: /var/svc/log/system-intrd:default.log
  Impact: This service is not running.
```

Refer to the knowledge article, SMF-8000-YX, for further instructions on fixing this problem.

## Repairing Faults or Defects

After Fault Management has faulted a component in your system, you will want to repair it. A repair can happen in one of two ways: implicitly or explicitly.

An *implicit repair* can occur when the faulty component is replaced or removed, provided the component has serial number information that the Fault Manager daemon can track. On many SPARC based systems, serial number information is included in the FMRIs so that the Fault Manager daemon can determine when components have been removed from operation, either through replacement or other means (for example, *blacklisting*). When such detections occur, the Fault Manager daemon no longer displays the affected resource in `fmadm fault` output. The resource is maintained in the daemon's internal resource cache until the fault event is 30 days old, at which point it is purged.

Implicit repairs do not apply to all systems. Sometimes, even though there is a chassis-id in the FMRIs, no FRU serial number information is available. So the Fault Manager daemon cannot detect a FRU replacement, requiring an *explicit repair*.

The `fmadm` command is used to explicitly mark a fault as repaired. Four syntaxes are associated with repairs for this command:

- `fmadm replaced fmri | label`
- `fmadm repaired fmri | label`
- `fmadm acquit fmri | label`
- `fmadm acquit uuid [fmri | label]`

Although these four commands can take FMRI and UUIDs as arguments, the preferred argument to use is the label. If a FRU has multiple faults against it, you want to replace the FRU only one time. If you issue the `fmadm replaced` command against the Label, the FRU is reflected as such in any outstanding cases.

## **fmadm replaced Command**

You can use the `fmadm replaced` command to indicate that the suspect FRU has been replaced or removed.

If the system automatically discovers that a FRU has been replaced (the serial number has changed), then this discovery is treated in the same way as if `fmadm replaced` had been typed on the command line. The `fmadm replaced` command is not allowed if `fmd` can automatically confirm that the FRU has not been replaced (the serial number has not changed).

If the system automatically discovers that a FRU has been removed but not replaced, then the current behavior is unchanged: The suspect is displayed as not present, but is not considered to be permanently removed until the fault event is 30 days old, at which point it is purged.

## **fmadm repaired Command**

You can use the `fmadm repaired` command when some physical repair has been carried out to resolve the problem, other than replacing a FRU. Examples of such repairs include reseating a card or straightening a bent pin.

## **fmadm acquit Command**

Often you use the `acquit` option when you determine that the resource was not the cause. Acquittal can also happen implicitly when additional error events occur, and the diagnosis gets refined.

Replacement takes precedence over repair, and both replacement and repair take precedence over acquittal. Thus, you can acquit a component and then subsequently repair it, but you cannot acquit a component that has already been repaired.

A case is considered repaired (moves into the `FMD_CASE_REPAIRED` state and a `list.repaired` event is generated) when either its UUID is acquitted, or all suspects have been either repaired, replaced, removed, or acquitted.

Usually `fmd` automatically acquits a suspect in a multi-element suspect list, or Support Services gives you instructions to perform a manual acquittal. You would only want to acquit by FMRI or label if you determined that the resource was not guilty in all current cases in which it is a suspect. However, to allow a FRU to be manually acquitted in one case while remaining a suspect in all others, the following option enables you to specify both UUID and FMRI, or UUID and label:

```
fmadm acquit uuid [fmri|label]
```

## Fault Management Log Files

The Fault Manager daemon, `fmd`, records information in several log files. The log files are stored in `/var/fm/fmd` and are viewed by using the `fmdump` command. See the [fmdump\(1M\)](#) man page for more information.

- The `errlog` log file records inbound telemetry information which consists of ereports.
- Informational events are recorded in two log files. `infolog_hival` is for high-value events, and `infolog` collects all other informational events.
- The `fltlog` log file records fault diagnosis and repair events.




---

**Caution** – Do not base administrative action on the contents of the log files, but rather on the `fmadm faulty` output. The log files can contain error statements, which should not be considered faults or defects.

---

The log files are automatically rotated. See the [logadm\(1M\)](#) man page for more information.

## Fault Statistics

The Fault Manager daemon, `fmd`, and many of its modules track statistics. The `fmstat` command reports those statistics. Without options, `fmstat` gives a high-level overview of the events, processing times, and memory usage of the loaded modules. For example:

```
# fmstat
module          ev_recv ev_acpt wait  svc_t  %w  %b  open solve  memsz  bufisz
cpumem-retire   1       0  0.0  403.5  0  0    0    0    419b   0
disk-transport  0       0  0.0  500.6  0  0    0    0    32b    0
eft             0       0  0.0   4.8   0  0    0    0    1.4M  43b
fmd-self-diagnosis 0       0  0.0   4.7   0  0    0    0     0     0
```

io-retire	0	0	0.0	4.5	0	0	0	0	0	0
snmp-trapgen	0	0	0.0	4.5	0	0	0	0	32b	0
sysevent-transport	0	0	0.0	1444.4	0	0	0	0	0	0
syslog-msgs	0	0	0.0	4.5	0	0	0	0	0	0
zfs-diagnosis	0	0	0.0	4.7	0	0	0	0	0	0
zfs-retire	0	0	0.0	4.5	0	0	0	0	0	0

The `fmstat(1M)` man page describes each column in this output. Note that the `open` and `solve` columns apply only to Fault Management cases, which are only created and solved by diagnosis engines. These columns are immaterial for other modules, such as response agents.

You may display statistics on an individual module by using the `-m module` option. This syntax is commonly used with the `-z` option to suppress zero-valued statistics. For example:

```
# fmstat -z -m cpumem-retire
NAME VALUE          DESCRIPTION
cpu_flts 1          cpu faults resolved
```

This example shows that the `cpumem-retire` response agent has successfully processed a request to take a CPU offline.



# Index

---

## A

acquit option, `fmadm` command, 62–63  
adding

- run control script (how to), 47
- SMF information, 21

`addpg` option, `svccfg` command, 21  
`addpropvalue` option, `svccfg` command, 21  
admin layer, description, 15–16  
administrative layers (SMF), *See* layers (SMF)  
all milestone

- description, 22
- example, 52–53

applying, profiles (SMF), 38  
ASR, FMA and, 57  
ASRU, definition, 55–56  
Auto Service Request, FMA and, 57  
Automatic System Reconfiguration Unit, *See* ASRU

## B

boot command

- all milestone, 52–53
- debug mode (`-m debug`), 53
- none milestone, 52–53
- verbose mode (`-m verbose`), 53

booting

- error logging (SMF) and, 22
- milestones (SMF) and, 22
- verbose mode, 53
- without starting services, 52–53

## C

changing, *See* modifying  
commands (SMF), list of, 18–19  
`config` property group, description, 20  
configuration repository (SMF), *See* repository  
configuring, services (SMF), 39–45  
converting

- `inetd` service to SME, 45
- run control script to SMF service, 48–49

corrupt repository, repairing, 50–52  
creating

- SMF manifest, 33–34
- SMF profiles
  - automatically, 36–37
  - manually, 37–38

customizations (SMF), displaying, 29

## D

debug log level, SMF, 17–18  
debugging, *See* troubleshooting  
default run level, definition, 23  
defects (FMA)

- displaying information about, 57–61
- notification of, 57
- repairing, 61–63

degraded SMF service state, description, 13  
`delcust` subcommand, `svccfg` command, 43–44  
`delcust` subcommand, `svccfg` command, 21  
delegated restarters, described, 19

- deleting
  - service customizations (SMF), 43–44
  - SMF information, 21
- delpropvalue subcommand, svccfg command, 21
- dependencies (SMF)
  - description, 12
  - displaying, 29–30
- dependents (SMF), displaying, 30
- determining, system's run level (how to), 24
- disabled service, troubleshooting, 50
- disabled SMF service state, description, 13
- disabling
  - run control script (how to), 48
  - service (SMF), 34
- displaying
  - customizations (SMF), 29
  - FMA information, 57–61
  - service status (SMF), 28–29

## E

- editprop subcommand, svccfg command, 40
- enabling, service (SMF), 35
- environment variables, modifying in SMF, 41–42
- errlog log file, 63
- error logging (SMF), description, 17–18
- /etc/init.d directory, 47
- /etc/inittab file, 25
  - entry description, 25
  - example of default, 25
- /etc/rc\*.d and, SMF, 22
- /etc/svc/profile/site profiles, 14–15

## F

- Fault Management Architecture, *See* FMA
- fault management resource identifier, *See* FMRI
- faults (FMA)
  - displaying information about, 57–61
  - notification of, 57
  - repairing, 61–63
- Field Replaceable Unit, *See* FRU
- fltl log file, 63

## FMA

- displaying information, 57–61
- fault statistics, 63–64
- log files, 63
- notification, 57
- notification parameters, 31–32
- overview, 55–56
- repairing faults or defects, 61–63
- fmadm command
  - example, 57–61
  - options, 62
  - overview, 61–63
- fmd daemon, overview, 55–56
- fmdump command
  - example, 59–60
  - FMA log files and, 63
- FMRI, description, 12–13
- fmstat command, example, 63–64
- FRU, definition, 55–56

## G

- general property group, description, 20

## I

- i option, svcbundle command, 34
- inetadm command
  - description, 18
  - example, 42–43
- inetd service
  - changing command-line arguments, 44–45
  - convert to SMF, 45
  - modifying property, 42–43
- infolog\_hival log file, 63
- infolog log file, 63
- init states, *See* run level
- initial snapshot, description, 17

## L

- layers (SMF), description, 15–16

legacy\_run SMF service state, description, 13  
 /lib/svc/bin/restore\_repository command,  
 example, 51  
 /lib/svc/manifest files, overview, 14  
 library interfaces, SMF, 19  
 listcust option, svccfg command, 20–21  
 listing, *See* displaying  
 listpg option, svccfg command, 20–21  
 listprop option, svccfg command, 20–21  
 logadm command, FMA and, 63

## M

maintenance SMF service state, description, 13  
 managing, services (SMF), 32–38  
 manifest (SMF), creating, 33–34  
 manifest layer, description, 15–16  
 manifests (SMF), description, 14  
 masking, SMF, 21  
 milestones (SMF)  
 booting and, 22  
 corresponding run level, 23  
 description, 11  
 modifying  
 command-line arguments, 44–45  
 multiple properties (SMF), 40  
 properties in inetd service, 42–43  
 service configuration files, 41  
 single properties (SMF), 39–40  
 SMF environment variables, 41–42  
 SMF information, 21  
 monitoring, services (SMF), 27–32  
 multiuser run level, description, 23

## N

new features, SMF, 9–10  
 none milestone  
 description, 22  
 example, 52–53  
 notification parameters  
 displaying, 29  
 setting, 31–32

## O

offline SMF service state, description, 13  
 online SMF service state, description, 13

## P

power-down run level, description, 23  
 profiles (SMF)  
 applying, 38  
 creating, 36–37  
 description, 14–15  
 properties (SMF)  
 description, 20  
 modifying multiple, 40  
 modifying single, 39–40  
 property groups (SMF), description, 20  
 psrinfo command, examples, 60

## Q

quiet log level, SMF, 17–18

## R

RBAC rights profiles, for SMF service, 33  
 reboot run level, description, 23  
 removing, *See* deleting  
 repaired option, fmadm command, 62  
 repairing  
 corrupt repository, 50–52  
 FMA faults or defects, 61–63  
 replaced option, fmadm command, 62  
 repository (SMF)  
 backups of, 17  
 description, 10, 15  
 restarter daemon (SMF), described, 19  
 restarter property group, description, 20  
 restarters (SMF), 19  
 description, 10  
 restarting, service (SMF), 35  
 restore\_repository command, example, 51  
 restoring, service (SMF), 36

- rights profiles, for SMF service, 33
- run control scripts
  - adding (how to), 47
  - converting to SMF service, 48–49
  - disabling (how to), 48
  - starting and stopping services, 46
  - tasks using, 45–49
- run levels
  - corresponding SMF milestone, 23
  - definition, 23
  - determining (how to), 24
  - multiuser with NFS
    - what happens when system is brought to, 26
- running snapshot, description, 17

## S

- service (SMF)
  - description, 11
  - enabling, 35
- service configuration repository, *See* repository
- service customizations (SMF), deleting, 43–44
- Service Management Facility, *See* SMF
- service states (SMF), description, 13
- service status, displaying, 28–29
- services (SMF)
  - configuration files
    - modifying, 41
  - configuring, 39–45
  - managing, 32–38
  - monitoring, 27–32
- setenv option, svccfg command, 21
- setting, notification parameters, 31–32
- showing, *See* displaying
- Simple Mail Transfer Protocol (SMTP), FMA and, 57
- Simple Network Management Protocol (SNMP), FMA and, 57
- single-user run level, description, 23
- site-profile layer, description, 15–16
- SMF
  - adding information, 21
  - booting and, 22
  - commands, 18–19
  - delegated restarters, 19

## SMF (*Continued*)

- deleting information, 21
- error logging, 17–18
- /etc/rc\*.d and, 22
- library interfaces, 19
- masking information, 21
- new features, 9–10
- notification parameters, 31
- overview, 9
- properties, 20
- RBAC rights profiles, 33
- restarter daemon, 19
- tasks, 27–54
  - troubleshooting, 49–54
  - viewing information, 20–21
- SMTP, FMA and, 57
- snapshots (SMF), description, 17
- SNMP, FMA and, 57
- start property group, description, 20
- start snapshot, description, 17
- starting and stopping services, 46
- svc.startd daemon, description, 19
- svcadm command, description, 18
- svcbundle command
  - creating manifests, 34
  - creating profiles, 37
  - description, 18
- svccfg command
  - delcust subcommand, 43–44
  - description, 18, 20–21
  - editprop subcommand, 40
- svccprop command
  - description, 18, 20–21
- svcs command
  - description, 18
  - example, 28
- system-profile layer, description, 15–16

## T

- tasks, SMF, 27–54
- troubleshooting
  - disabled service, 50
  - SMF service, 49–54

**U**

uninitialized SMF service state, description, 13  
unsetenv subcommand, svccfg command, 21  
/usr/bin/svcprop command, description, 18  
/usr/bin/svcs command  
    description, 18  
    example, 28  
/usr/lib/fm/fmd/fmd daemon, overview, 55–56  
/usr/sbin/fmadm command, example, 57–61  
/usr/sbin/fmdump command  
    example, 59–60  
    FMA log files and, 63  
/usr/sbin/fmstat command, example, 63–64  
/usr/sbin/inetadm command, description, 18  
/usr/sbin/logadm command, FMA and, 63  
/usr/sbin/psrinfo command, examples, 60  
/usr/sbin/svcadm command  
    description, 18  
/usr/sbin/svccfg command  
    description, 18, 20–21

**V**

/var/fm/fmd log files, 63  
/var/svc/manifest files, overview, 14  
verbose log level, SMF, 17–18  
viewing, SMF information, 20–21

**W**

who command, 24

